

NASA-CR-174,892

CR-174892



National Aeronautics and
Space Administration

NASA-CR-174892
19860012461

ato
15822

BURNER LINER THERMAL/STRUCTURAL LOAD MODELING

By
R. Maffeo

**General Electric Company
Aircraft Engine Business Group
Cincinnati, Ohio 45215**

**Prepared For
National Aeronautics and Space Administration
Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135**

LIBRARY COPY

MAY 28 1986

**NASA-Lewis Research Center
Contract NAS3-23272**

**LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA**



NF01214

1. Report No. CR-174892		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Burner Liner Thermal/Structural Load Modeling				5. Report Date October 1984	
				6. Performing Organization Code	
7. Author(s) R. Maffeo				8. Performing Organization Report No.	
				10. Work Unit No.	
9. Performing Organization Name and Address General Electric Company Aircraft Engine Business Group Cincinnati, OH 45215				11. Contract or Grant No. NAS3-23272	
				13. Type of Report and Period Covered Final Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, OH 44135				14. Sponsoring Agency Code	
15. Supplementary Notes Project Manager, Dr. R.L. Thompson Structures and Mechanical Technologies Division NASA Lewis Research Center, Cleveland, OH 44135					
16. Abstract <p>The software package developed under this contract was called <u>Transfer Analysis Code to Interface Thermal/Structural Problems (TRANCITS)</u>. The TRANCITS code satisfies all of the objectives for transferring thermal data between heat transfer and structural models of combustor liners; in addition, it can be used as a generic thermal translator between heat transfer and stress models of any component, regardless of the geometry. TRANCITS can accurately and efficiently convert the temperature distributions predicted by the heat transfer programs to those required by the stress codes. It can be used for both linear and nonlinear structural codes and can produce nodal temperatures, elemental centroid temperatures, or elemental Gauss point temperatures. The thermal output of both the MARC and SINDA heat transfer codes can be interfaced directly with TRANCITS and it will automatically produce stress model codes formatted for NASTRAN and MARC. In addition to these codes, any thermal program and structural program can be interfaced by using the neutral input and output forms supported by TRANCITS.</p> <p>In summary, the TRANCITS code can be used to interface temperature data between thermal and structural analytical models. The use of this transfer module allows the heat transfer analyst to select the thermal mesh density and thermal analysis code best suited to solve the thermal problem and gives the same freedoms to the stress analyst, without the efficiency penalties associated with common meshes and the accuracy penalties associated with the manual transfer of thermal data.</p> <p><u>KEY WORDS</u></p> <p>Interface, Heat Transfer Analysis, Structural Analysis, Finite Element, Iso-parametric, Finite Difference, Transfer Module</p>					
17. Key Words (Suggested by Author(s)) Heat Transfer Stress Analysis Computer-Aided Engineering			18. Distribution Statement		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 203	
				22. Price*	

* For sale by the National Technical Information Service, Springfield, Virginia 22161

N86-21932#

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1.0 SUMMARY	1
2.0 INTRODUCTION	2
3.0 DESCRIPTION OF TASKS	5
3.1 Task I - Current Status of Thermal Load Modeling	5
3.2 Task II - Assessment of Thermal Load Modeling	5
3.3 Task III - Development of Thermal Load Transfer Module	6
3.4 Task IV - Verification and Documentation of Automated Automated Thermal Transfer Load Module	6
4.0 TASK I	8
4.1 Thermal Analysis Programs	8
4.2 Structural Analysis Codes	9
5.0 TASK II	11
5.1 2D Titan Evaluation	11
5.2 Merlin Evaluation	12
5.3 Lomap Evaluation	12
5.4 Evaluation of Transferring Criteria	13
6.0 DEVELOPMENT AND DOCUMENTATION OF THE TRANSFER MODULE	17
6.1 General Flow of Code	17
6.2 Input and Output	19
6.3 Finite Difference Consideration	21
6.4 3D Search and Weighting Coefficient Routines	24
6.5 Exterior Point Surfacing Routine	31
7.0 VERIFICATION AND DOCUMENTATION OF THE TRANSFER MODULE	34
7.1 Verification of the Transfer Module	34
7.2 Documentation of the Transfer	39
8.0 CONCLUSIONS	51
APPENDIX A	53
APPENDIX B	66
APPENDIX C	75
APPENDIX D	180
REFERENCES	203

1.0 SUMMARY

The objective of this program was to develop a thermal load transfer module that would be capable of transferring thermal data from a three dimensional analytical model of a combustor liner to a corresponding three dimensional stress analysis model. The initial goal of the transfer module was the ability to handle different mesh densities for the heat transfer analysis and the stress analysis and the capability to utilize both finite difference and finite element heat transfer codes. The extended goals of the transfer module were to develop a general purpose tool that would effectively allow for the coupling of heat transfer analysis and stress analysis while maintaining the flexibility of independent analysis codes.

The software package developed under this contract is called Transfer Analysis Code to Interface Thermal/Structural Problems or TRANCITS. The TRANCITS code satisfies all of the objectives for transferring thermal data between heat transfer and structural models of combustor liners and, in addition, can be used as a generic thermal translator between heat transfer and stress models of any components regardless of the geometry. TRANCITS can accurately and efficiently convert the temperature distributions predicted by the heat transfer programs to those required by the stress codes. It can be used for both linear and non-linear structural codes and can produce nodal temperatures, elemental centroid temperatures or elemental gauss point temperatures. The thermal output of both the MARC and SINDA heat transfer codes can be interfaced directly with TRANCITS and it will automatically produce stress model temperatures formatted for NASTRAN and MARC. In addition to these codes, any thermal program and structural program can be interfaced by utilizing the neutral input and output forms supported by TRANCITS.

In summary, the TRANCITS code can be used to interface temperature data between thermal and structural analytical models. The use of this transfer module allows the heat transfer analyst to select the thermal mesh density and thermal analysis code that is best suited to solve the thermal problem and gives the same freedoms to the stress analyst without the efficiency penalties associated with common meshes and the accuracy penalties associated with the manual transfer of thermal data.

2.0 INTRODUCTION

The overall objective of this program was to develop a thermal data transfer computer program module for the Burner Liner Thermal-Structural Load Modeling Program. This was accomplished by (1) reviewing existing methodologies for thermal data transfer and selecting three heat transfer codes for application in this program, (2) evaluating the selected codes to develop criteria for developing a computer program module to transfer thermal data from the heat transfer codes to selected stress analysis codes, (3) developing the automated thermal load transfer module, and (4) verifying and documenting the module.

In aircraft turbine engine hot section components, combustor liners, and turbine blades and vanes, cyclic thermal stresses are the most important damage mechanism. Consequently, accurate and reliable prediction of thermal loads is essential to improving durability. To achieve this goal, a considerable effort over the past 20 years has been devoted to the acquisition of engine temperature test data, as well as the development of accurate, reliable, and efficient computer codes for the prediction of steady-state and transient temperatures and for the calculation of elastic and inelastic cyclic stresses and strains in hot section components. There is a need for continued development of these codes, because the availability of more accurate analysis techniques for complex configurations has enabled engine designers to use more sophisticated designs to achieve higher cycle efficiency and reduce weight.

It has become apparent in recent years that there is a serious problem of interfacing the output temperatures and temperature gradients from the heat transfer codes with the input to the stress analysis codes. In part, this is a penalty for success. When computers were slower and computer memory smaller, the size of problems that could be analyzed, in terms of heat transfer nodes and stress analysis finite elements, was limited. Manual transfer of the output temperatures from the heat transfer codes to the stress analysis input, with manual interpolation to accommodate the mismatch in the nodal meshes of the two programs, was not unduly burdensome. With the growth in computer capacity and speed and the development of input preprocessors and output postprocessors, the analysis of components using hundreds and even thousands of nodes in the heat transfer and stress models has become economical and routine. This has exacerbated the interfacing problem to where the engineering effort required is comparable to that required for the remainder of the analysis. Furthermore, a considerable amount of approximation has been introduced in an effort to accelerate the process. This tends to introduce errors into the temperature data which negates the improved accuracy in the temperature distribution achieved through use of a finer mesh.

One solution to these interfacing problems is to force the heat transfer model and the stress model to be performed using the same analysis code and the exact same mesh densities. Although this technique may be acceptable for some applications, it has several disadvantages. In many models the areas of high stress gradient may be influenced by thermal loads but are predominantly

controlled by geometrical discontinuities such as fillets and holes. A structural mesh that was fine enough to represent the gradients due to these discontinuities would be in general much denser than necessary for the thermal problem. This "overkill" might be tolerated for 2D analysis but many times the additional cost associated with these extra nodes in 3D analysis cannot be afforded. It is sometimes argued that because of the reduced degrees of freedom for a thermal problem as compared to a structural problem (typically 1 dof for thermal and 2 to 6 dofs for structural) that this additional cost is minimized. This is true for steady-state analyses, but for transient problems, the number of thermal runs far exceeds the number of structural runs usually performed.

Another example of the disadvantage of common meshes is the current trend in structural analysis toward adaptive mesh refinement. This technique will automatically refine the structural mesh in areas of high strain gradients. Unless the high strain gradient were caused by high thermal gradients there is certainly no need to refine the thermal mesh and redo the thermal analysis just because the structural mesh was made denser. All of these factors point toward the need for different thermal and structural mesh densities.

A further drawback of common heat transfer and stress codes and meshes is the widespread use of the finite difference technique to solve the thermal problem as opposed to the use of finite element codes for the structural problem. Again this suggests the need for an external process to transfer the thermal data to an independent structural analysis.

Therefore, the primary objective to this project is to develop, implement and demonstrate an interfacing module that will address the problems associated with the transfer of temperature data from thermal analyses to structural analyses.

This interfacing module must be able to handle different thermal and structural meshes and must support both finite difference and finite element heat transfer codes.

Conceptually, the overall program structure is shown in Figure 1. Direct inputs are temperature data, and the geometries associated with the thermal data and with the stress model. The particular heat transfer and stress analysis programs used affect the process inside the transfer module. Final output is the temperature data at the required locations in the stress model.

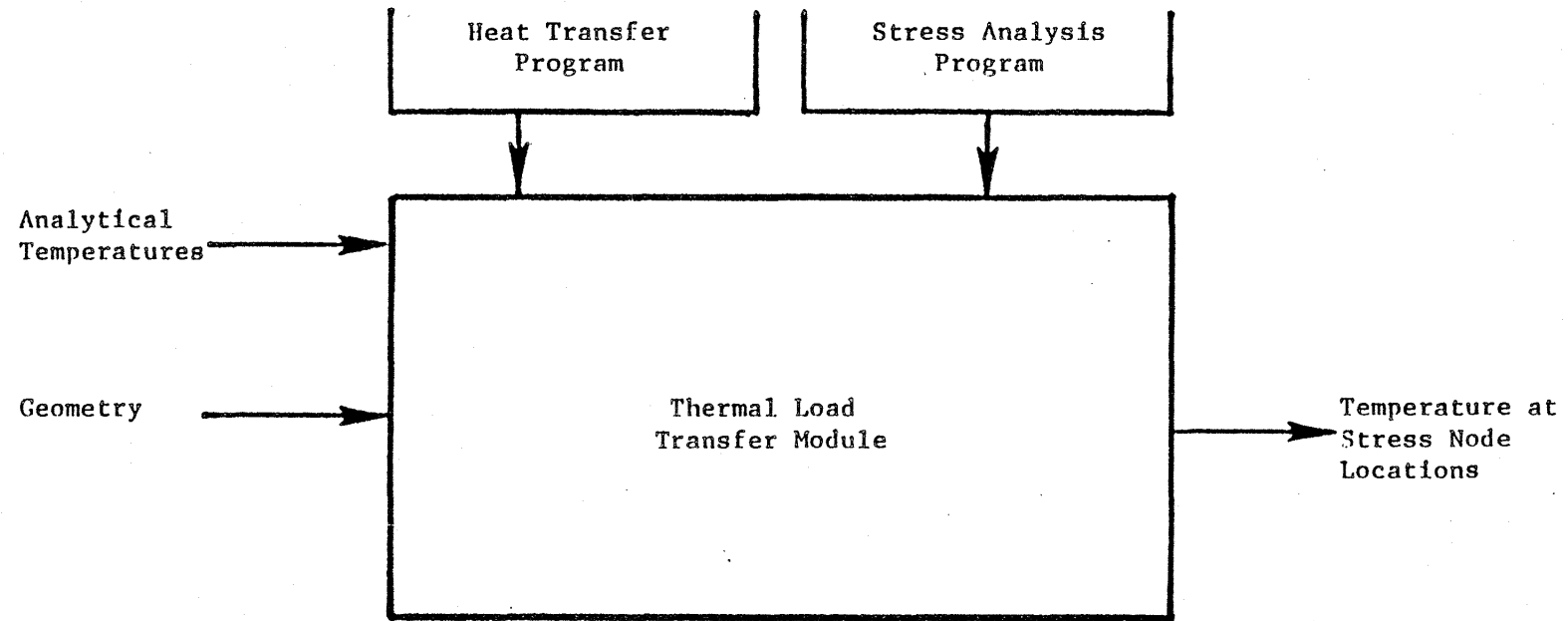


Figure 1. Overall Program Structure.

3.0 DESCRIPTION OF TASKS

This project was addressed by completing four tasks. These tasks include a review of the current status of thermal load modeling, an assessment of the thermal load modeling procedures, development of a thermal load transfer module and verification and documentation of the transfer module.

The four tasks are described below.

3.1 TASK I - CURRENT STATUS OF THERMAL LOAD MODELING

General Electric conducted a review of the current methods used to transfer thermal data from analytical sources to structural analysis programs in order to compute thermal stress and strain. The review included both in-house capabilities and papers in the open literature. The feasibility of the available methods were assessed and incorporated into the thermal load transfer module when applicable. In addition to reviewing the existing transfer techniques, General Electric also conducted a review of existing numerical temperature prediction codes including both in-house programs and those available in the open literature. This review was conducted with particular attention paid to the existing thermal load transfer methodologies and also included an examination of the temperature output format for each code reviewed. Based upon the results of this review, three computer codes were identified from among those available to the public to be used in the verification of the thermal load transfer module which was developed in Task III of this contract.

A review was also made of structural analysis computer programs as the first step in the process of selecting those codes to be used in this program. Both in-house computer programs and those programs defined in the open literature were reviewed. The objective of this review was to produce an evaluation of each code's input-output formulation, particularly with respect to temperature information. For the greatest public utility, primary emphasis was placed on those codes which are both in the open literature and actively supported by the commercial computer houses.

3.2 TASK II - ASSESSMENT OF THERMAL LOAD MODELING

In Task II, the Aircraft Engine Business Group outlined the thermal data transfer procedures for each of the methodologies approved in Task I and identified, discussed, and assessed their limitations and deficiencies in order to clearly define and evaluate the problem areas associated with thermal load modeling. Particular emphasis was placed on the questions of compatibility, ease of implementation, and thermal data format in the discussion of the current deficiencies and limitations involved in the transfer of thermal data from heat transfer analysis to the various structural analysis codes.

Once the limitations and deficiencies were identified in this task, a consistent set of evaluation criteria was developed. These criteria included factors that address the accuracy and the efficiency of the transfer procedure.

3.3 TASK III - DEVELOPMENT OF THERMAL LOAD TRANSFER MODULE

Based on the results of Tasks I and II, General Electric developed a thermal data transfer module that efficiently and accurately interfaced thermal data into structural analysis codes. The thermal data may come from both finite difference and finite element heat transfer analysis programs. The transfer module is capable of interfacing 3-D steady-state and transient heat transfer data into linear and nonlinear structural analysis programs.

State-of-the-art features in this module allow different mesh distribution for the heat transfer model, and the structural analysis model and slight differences in part geometry due to differences in component tolerances used in the models. Three-dimensional translation and rotation routines were incorporated into the module to account for possible mismatches between the origin of the structural model and the thermal model coordinate systems.

As these techniques were developed, they were encoded into a modular software package that efficiently manages the thermal data. The temperature data from the selected heat transfer programs is read directly by the interfacing module and automatically converted into a usable form for the selected structural analysis codes. The output of the interfacing module is completely compatible with the requirements for all of the structural codes selected. In addition, the flexible format of the output allows it to be easily implemented into other structural codes.

The techniques developed for the transfer module were designed to be flexible and have as much built-in generic capability as possible. The module provides options for compatible input with different existing heat transfer and structural computer codes. In conclusion, the transfer module provides an accurate streamline method of determining burner liner temperatures and gradients at specified locations compatible with the structural analysis input requirements.

3.4 TASK IV - VERIFICATION AND DOCUMENTATION OF AUTOMATED THERMAL TRANSFER LOAD MODULE

This Task is concerned with validating the data transfer module which was developed in Task III. Several test cases were run utilizing various combinations of heat transfer and structural analysis computer codes with temperature data flowing between the programs by means of the data transfer module. One of the test cases involved a 3D model of a combustor liner. Both finite difference and finite element heat transfer codes were used to produce the thermal results and these thermal loads were then interfaced to structural analysis codes using the transfer module. Cases were run using similar and dissimilar meshes for the heat transfer and stress analysis. Stress points that lie slightly outside the heat transfer model were interfaced using the

module. The accuracy of the transfer was evaluated by comparing the temperatures produced by the transfer module to those predicted by the actual heat transfer codes used.

These demonstration test cases were documented as verification of the transfer code.

The coding of the transfer module was made as hardware independent as possible. All computer system dependent routines were isolated into easy to modify subroutines. The final version of the transfer module is executable on NASA Lewis's IBM system.

As a separate document a User's Manual for the data transfer module was provided. This manual includes a precise set of operating instructions for each of the software packages developed. Included is a description of all variables needed to execute the codes.

As part of this documentation, a complete set of Fortran listings for each subroutine developed for the data transfer module along with a computer tape of this coding was delivered to NASA Lewis Research Center.

4.0 TASK I

Current status of thermal load modelling and a review of heat transfer and structural codes.

4.1 THERMAL ANALYSIS PROGRAMS

A review was conducted of the in-house and open literature concerning numerical temperature prediction codes. The primary purpose of this task was to select three codes to be used to verify the thermal load transfer module. The other criterion for the codes is that they must be available in the NASA Lewis computer system.

A comprehensive summary of the capabilities of thirty-eight different codes is contained in the excellent survey article by A.R. Noor entitled "Survey of Computer Programs for Heat Transfer Analysis". (Reference 1) We studied this summary closely, paying particular attention to any already existing transfer modules between the thermal and structural analysis modules, as well as to the output form for the temperature results.

As to the first question, none of the programs surveyed contained an automatic thermal interface. This statement is also applicable to all our in-house heat transfer programs as well. Most major finite element programs do, however, have the capability to pass thermal data directly to an identical mesh which can then be used for structural analysis. Such a capability may be extremely useful for certain applications, but for many applications it is very restrictive. It imposes mutual restrictions on the heat transfer analyst and on the structural analyst which neither of them really need to observe. It is also true that this one-to-one correspondence requires that the heat transfer program use a finite element technique. This fact is clearly not a generally desirable arrangement.

The second area we focused on was how we could access the temperatures produced by the heat transfer code. Most of the major codes surveyed also carry a file of output temperatures, which are suitable for post-processing. The form of this file is, of course, program dependent, but we believe that it can become the basic input to the thermal loads interface module. A thermal output file is not absolutely necessary since all the thermal codes print out the temperature at the heat transfer elements in some readable form. This readable form, known as "hard-copy", can be scanned and the required thermal data can be obtained.

Since all the programs considered possess both the special output file as well as the standard "hard copy" output, either form of the output could be utilized. One of the unknown questions is: Does the special output file contain all the information necessary for the thermal load interface? Frequently finite difference based programs, for example, do not carry a complete geometry definition as a part of the output. The complete geometry is essential to the interface module. It can, in most cases, be obtained from the

input to the heat transfer program. Questions such as this one cannot be answered with confidence until the individual programs are actually tried.

In general, all programs were found to be roughly equivalent as far as producing temperature data usable by a structural program. Many different features are available in these codes, for example, different treatments of radiation, better plotting capabilities, etc., but the main focus of this study is on transferring temperature data. On this basis, there are nine major programs which could be used:

- ANSYS
- HEATING
- MARC
- NASTRAN
- SINDA
- THTD
- ADINAT
- SAHARA
- PAFEC

Of these codes, only four are currently available on NASA Lewis computers and these are HEATING, MARC, NASTRAN, and SINDA. Further, the transfer module is demonstrated with the interface of both finite difference and finite element codes, namely the SINDA program for the finite difference program and MARC as the finite element program. Also demonstrated is the interface to NASTRAN, because of its wide popularity. For completeness, the portions of Noor's survey article which deal with these programs and others are listed in Appendix A.

Other codes considered, but not available on NASA Lewis's computer, were ANSYS, ADINAT, SAHARA and PAFEC. ANSYS and ADINAT are both widely used finite element codes. Both of these codes have a wide range of elements that can be used for heat transfer analysis, and ANSYS is frequently used for gas turbine design problems. It would be desirable to have the transfer module deal with these heat transfer codes directly. The SAHARA program developed at Sandia laboratories is another finite difference program that could be used to validate the transfer module. The survey indicates that SAHARA, like SINDA, has an output file that can be used for postprocessing. This output file would be very useful for the transfer module. Finally the PAFEC program was considered because it uses boundary integral techniques to perform the heat transfer analysis. Here again, the ability to transfer thermal data from such a program is beyond the requirements of this contract but nevertheless will be very desirable. It is felt, however, that because of the modular construction of the transfer module, future implementation of these codes or any other common code could be easily accomplished.

4.2 STRUCTURAL ANALYSIS CODES

A survey of stress analysis programs was also conducted. By virtue of the huge number of programs available (it is certainly in the hundreds), our survey cannot be considered comprehensive, but our conclusions are consistent

with the scope of this study. In particular, no structures program which was considered carries its own thermal interface transfer module. Temperature data are transferred by one of two methods.

- (a) Temperatures are obtained from an exactly equivalent heat transfer analysis, utilizing the identical mesh,

or

- (b) temperatures are interpolated and/or extrapolated from a different mesh to discrete points: nodes, gauss points, element centroids, etc.

Method (a) requires no interface module whatsoever, but it is not considered to be desirable for a general case.

Method (b) is ideally suited for a thermal interface module in that it is quite general. The module which was developed in Task III produces the temperature at any point in space which is consistent with a specific stress analysis. Whether this point is an actual node or gauss point makes little difference. The actual details of which points are required for temperature input to a structural model are not critical. The module will accept any selection.

Our survey of commercially available programs relied heavily on a recent review with R. Zirin of the General Electric Gas Turbine Division covering ANSYS, EASE2, MARC, NASTRAN, STARDYNE, and SUPERB.

All of these finite element structural codes have a large library of different elements. The location within these elements to which the thermal data must be input varies. Some require nodal temperatures, some need elemental centroid temperatures, others need gauss point data. They all, however, need temperatures at some discrete point in the structural model. Once again, the transfer module will produce temperatures at any location.

For the purposes of this contract, the transfer module is demonstrated on MARC and NASTRAN, the codes available at NASA Lewis. Again, for completeness, the information on MARC and NASTRAN is given in Appendix B.

The thermal load interface module developed under this study is flexible enough to drive any structural program which needs temperatures for input.

5.0 TASK II

Assessment of Thermal Load Modeling

The objective of Task II was to evaluate the thermal transferring capabilities currently available in existing heat transfer codes. In addition, an attempt was made to identify any freestanding thermal data transfer modules that might be in use by industry. Once these capabilities were evaluated, a set of criteria was compiled that included these factors and several additional criteria that were important in the development of a successful transfer module.

As stated in Section 4.1 no general capability for transferring 3D thermal data to structural codes was identified in the thermal codes surveyed. The most common mechanism for accomplishing this interfacing was through the use of similar mesh densities. One exception to this was the THTD program that did produce a thermal output file that could input temperatures directly into selected stress codes. The actual procedure of assigning the stress node temperatures, however, involved a manual correlation between the stress point and the heat transfer element chosen to be "closest" to the stress node. Not only was this time consuming, but it also did not provide for any interpolation between heat transfer elements. The proposed code would consider this shortcoming in its development.

The investigation of external thermal load transfer modules identified three codes being used by various companies. These codes were 2D TITAN developed by General Electric (AEBG), MERLIN developed by Sandia Lab and LOMAP developed by Lockheed. The evaluation of each of these codes follows:

5.1 2D TITAN EVALUATION

The 2D TITAN transfer code was developed by GE to interface the thermal results of a heat transfer code called THTD to selected structural analysis codes. The THTD code is a finite difference formulation. The transfer code will interface temperatures into any 2D mesh including plane stress, plane strain, axisymmetric and shells of revolution. It has some heat transfer windowing capability both spatial and temporal. TITAN can interface stress points that lie slightly outside of the heat transfer model. It also has built in 2D coordinate transformations to align the thermal and structural models. The program has some internal accuracy checks used to flag areas of the thermal model where the interfacing could yield inaccurate results. Another related feature is the use of interpolating functions that are consistent with the assumption associated with the THTD heat transfer element. The actual interpolation is accomplished using the 2D planar isoparametric shape functions.

The following limitations were identified:

- a. No 3D capability.
- b. Only interfaces with one heat transfer code.

- c. No time interpolations between thermal solutions.
- d. No provision for scaling the thermal results based on variations in thermal boundary conditions.

5.2 MERLIN EVALUATION

The Sandia code (MERLIN) is very similar to the in-house GE program called TITAN. Investigation into MERLIN revealed the following limitations:

- a. Restricted to 2D problems.
- b. Interfaces only with finite element heat transfer codes.
- c. Does not interface with finite difference heat transfer codes.
- d. Only interfaces into a specific set of finite element stress codes.
- e. Meshes (both heat transfer and stress) must be generated using a specific mesh generator.
- f. Does not allow for coordinate transformation of meshes.
- g. Does not account for stress points "slightly" outside heat transfer mesh.
- h. Limits the number of elements in the heat transfer model.
- i. Limits the number of nodes in stress model.
- j. Uses the same temperature mapping function regardless of the type of heat transfer element.

The interfacing technique used in MERLIN is a three-step process. Step one is the mesh search. This search locates which heat transfer element contains each stress point. The second step is called the element search. This process defines the local coordinates of the stress point with respect to the coordinates of the heat transfer element. The last step computes the temperature of each stress point based on the weighting coefficients derived from the local coordinates. This entire process is almost identical to the methods used in TITAN.

5.3 LOMAP EVALUATION

The Lockheed program LOMAP has very limited capabilities as a general transfer module. It will transfer temperatures from a 3D heat transfer mesh to a 3D stress mesh, however, the method used is very simplistic. The weighting coefficients used to interpolate temperatures are Proportional to the temperatures of the four closest heat transfer nodes. For a general transfer program

the four closest nodes may not be the correct nodes to use. In fact, the four geometrically closest heat transfer nodes may not even be in the same thermal region as the stress point. This technique places severe restriction on the types of meshes that can be interfaced properly.

5.4 EVALUATION OF TRANSFERRING CRITERIA

Based on the study of existing thermal transfer modules, GE internal needs, and GE previous experience with TITAN, three levels of criteria for the program development were proposed.

Level I contains the general criteria which must be satisfied for a usable product.

Level II contains specific criteria which must be satisfied to meet the requirements associated with gas turbine design problems. This list stems mainly from GE internal experience.

Level III contains criteria which are desirable but not necessary. In most cases, items in this class can be achieved through a multi-step Process. Total automation might be desirable, but we do not believe this effort is warranted at this time.

Level I: General Criteria For A Thermal Transfer Module

IA. Independent Heat Transfer and Stress Geometry Meshes

This criteria lies at the heart of our effort. Useful thermal transfer modules must address this feature. Automatically included in this is the ability to transfer from finite difference heat transfer mesh to a finite element stress analysis mesh.

IB. Accurate Transfer of Data

Simplistic approaches such as averaging the closest nodes do not always yield accurate results, and the utility of the transfer program is then questionable. This criteria was met by using all available temperature information to do the interpolation and by using different mappings to correspond with different heat transfer elements.

IC. User Friendly

Programs which do not meet this criteria tend to be used incorrectly or as a last resort. The plan was to construct the thermal transfer module to encourage the analyst to use it. Any errors encountered by the module are flagged and reported with a diagnostic message.

ID. Computationally Efficient

The program was coded to achieve an efficient flow of data. GE's past experience with TITAN has led to several improvements over the original efforts, and these have resulted in similar gains in 3D transfer problems. This criteria covers both searches to find the proper heat transfer element for a stress node as well as the single element inverse mapping functions.

IE. Flexible

The thermal transfer module was constructed such that future modifications, or even different applications, for instance, pressure or boundary condition transfer, could be accomplished without a full rewrite. This criteria stems from past experience in having to improve or draw upon old techniques which could almost, but not quite, perform the required task. The transfer module not only transfers temperatures in a state-of-the-art manner, but it will also provide a vehicle for other 3D interpolation based problems.

Level II - Specific Criteria Required For Gas Turbine Design Problem

IIA. Coordinate Transforms

Coordinate transformation that allow the heat transfer model to be aligned with the structural model.

IIB. "Out-of-Box" Provision

Provision to account for stress nodes that lie just outside the heat transfer model due to slight differences in the dimensions used in the heat transfer and stress analysis models, as a result of using different tolerances on the actual component dimensions.

IIC. Windowing

Capability to "window in" on a smaller portion of the heat transfer model.

IID. Selected Time Steps

Ability to select temperature distributions at specific time steps form a large transient thermal analysis.

Level III - Desirable But Not Essential Features

IIIA. Automatic Handling of Temperature Discontinuities

In the module these will be treated in a two-step manner. Total automation is possible, but has not been implemented.

IIIB. Scaling of Temperatures Based on Variation In Engine Power Level Settings

Scaling of the interpolated temperatures based on changes in the heat transfer boundary condition, such as engine power settings, various inlet conditions or different cooling flows requires specific knowledge of the heat transfer analysis and will not be done inside the transfer module, but could, if desired, be applied by another program to the original results from the transfer module.

IIIC. "Altered" Stress Geometry

Many times the stress analyst wants to alter the part geometry to reduce his stresses, but the deviations will not, in the judgement of the heat transfer analyst, affect the temperatures. In the past, "ad hoc" procedures to transfer temperatures to the new stress geometry have been used. This approach is not optimum and has not been included in the transfer module.

Table I compares each of the external transfer codes evaluated against some of the criteria chosen. This table illustrates areas where significant effort was placed to develop an accurate and efficient thermal transfer module.

Table 1.

Criteria	MERLIN	Codes LOMAP	GE TITAN	TRANCITS Proposed Transfer Module
Independent Meshes	Yes	Yes	Yes	Yes
2-D Capability	Yes	Yes	Yes	Yes
3-D Capability Thermal Mapping a Function of Heat Transfer Elements	No	Yes	No	Yes
Coordinate Transformations	No	No	Yes	Yes
'Out-of-Box'	No	No	Yes	Yes
Provision Isoparametric	Yes	No	Yes	Yes
Mapping Geometrical/Temporal	No	No	Yes	Yes
Windowing Capability For Time Interpolation	No	Yes	No	No

6.0 DEVELOPMENT AND DOCUMENTATION OF THE TRANSFER MODULE

The objectives in this task were to develop an efficient and accurate software package that implemented all of the features discussed in Task I and Task II. The development was divided into several subtasks, each of which addressed important program features that were critical to a successful code.

The subtasks were:

1. General Flow of Code
2. User Input
3. General Heat Transfer Input
4. Specific Heat Transfer Code Input
5. General Stress Code Input
6. General Output
7. Specific Stress Code Output
8. Finite Difference Heat Transfer Code Consideration
9. Finite Element Heat Transfer Code Consideration
10. General 3D Search Routines
11. Weighting Coefficient Routines
12. Exterior Surfacing Routines

6.1 GENERAL FLOW OF CODE

Several factors were considered in developing the general flow of the code. These factors included:

- a. Program modularity
- b. Program efficiency
- c. Input/output requirements
- d. Internal data flow
- e. Flexibility/portability
- f. User friendliness

The basic program flow is illustrated by Figure 6-1.

Each of the boxes in Figure 6-1 represent program modules that have been coded to minimize the system dependent features of the transfer program. In addition to this, these modules communicate with each other through an internal file structure that is hardware independent, easy to follow and easy to modify.

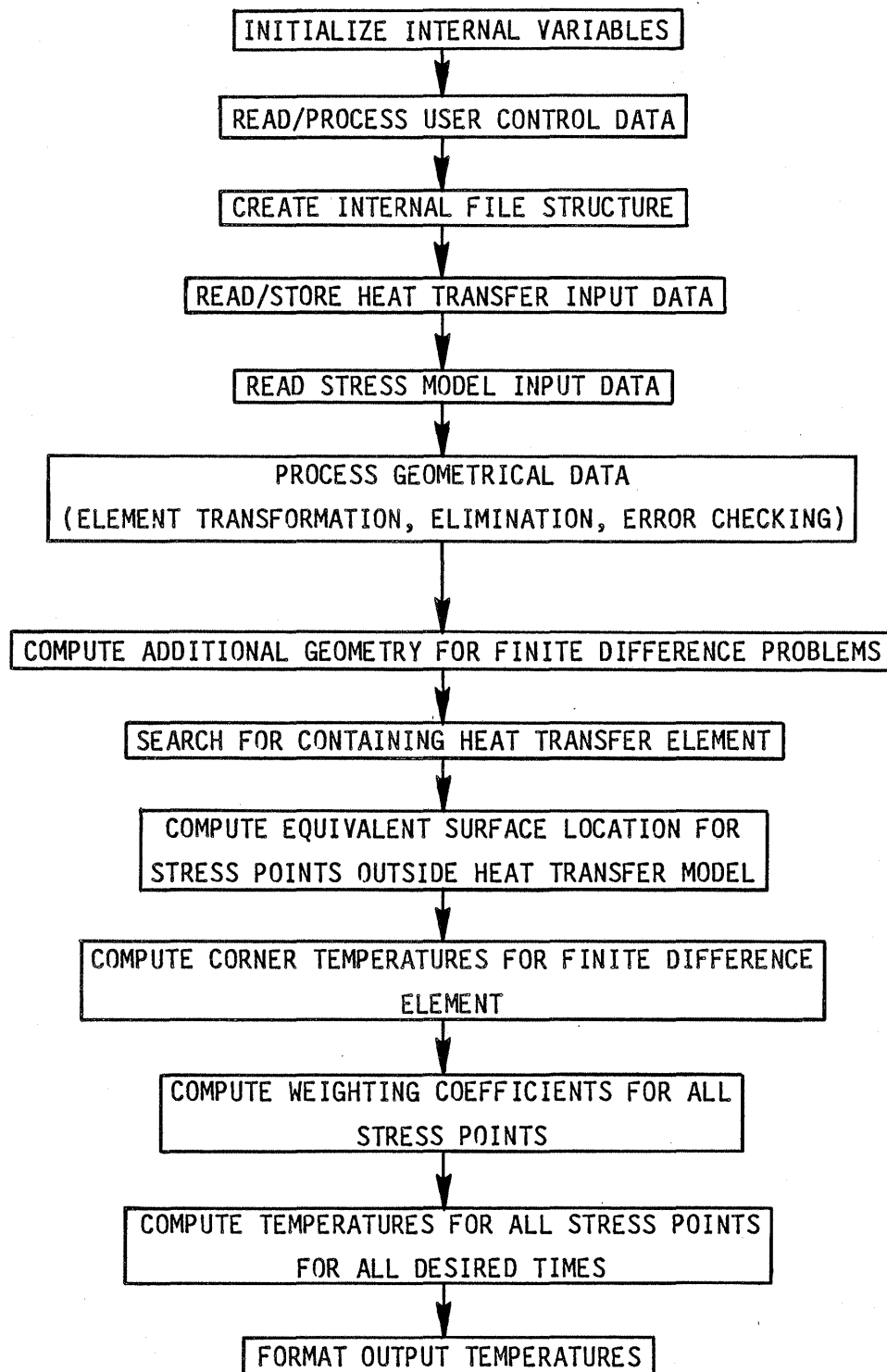


Figure 6-1. Overall Modular Flow of Code

These features not only make the code very portable but also allow the same basic structure to be used in the future to transfer other types of loading. Additional heat transfer elements can easily be added to this scheme.

6.2 INPUT AND OUTPUT

The input and output information can be grouped into three basic categories. They are geometrical, thermal and user control data. Figure 6-2 illustrates how these data interact with the transfer module.

The user control data are input to the transfer module with a namelist input file. This file contains all of the variables that can be input by the user to control the interfacing procedure and is described in detail in the TRANCITS Users manual (Reference 2). Options such as geometry and temporal windowing, coordinate transformations, time step tolerancing and information about the type of input and type of output desired are entered through this file and help provide for the user-friendliness of the code.

The other two categories of input are associated with the geometrical data for both the heat transfer and stress models and the temperature data that is the output from a thermal code. The transfer module has been structured to accept output of the SINDA finite difference code and the thermal output of the MARC finite element program. In order to satisfy these requirements and at the same time produce a module that can be used with a variety of heat transfer codes, a neutral heat transfer input form was developed. This input form will support both finite difference and finite element thermal results and is the primary input to TRANCITS. Compatibility with SINDA and MARC is accomplished by internally converting the output of these codes to this neutral form.

The heat transfer input file consists of six partitions of data. These data are described in detail in the users manual. It stores all the sizing, identification, geometrical and thermal data associated with the heat transfer analysis. It is important to note that the geometrical data is an essential part of this input. The geometry (nodal coordinates and element connectivity) is the only link between the heat transfer and stress models. Without geometrical data an accurate interface cannot be accomplished. The thermal data in this file varies depending on the type of heat transfer analysis. Nodal temperatures are stored if the thermal code is a finite element type and elemental temperatures are stored if the thermal code is a finite difference formulation.

The automatic interface with SINDA and MARC makes use of output files created by these programs. The MARC program produces an output file known as the MARC cost file. This output contains all of the data required by the transfer module to perform the interface. The SINDA code also has the capability to produce an output file. However, since SINDA is a finite difference program, there is no geometry stored on the output file. Indeed, the SINDA code does not require geometry as input. Only volumes, areas and distances are needed to predict the thermal distributions. Therefore, the raw SINDA output

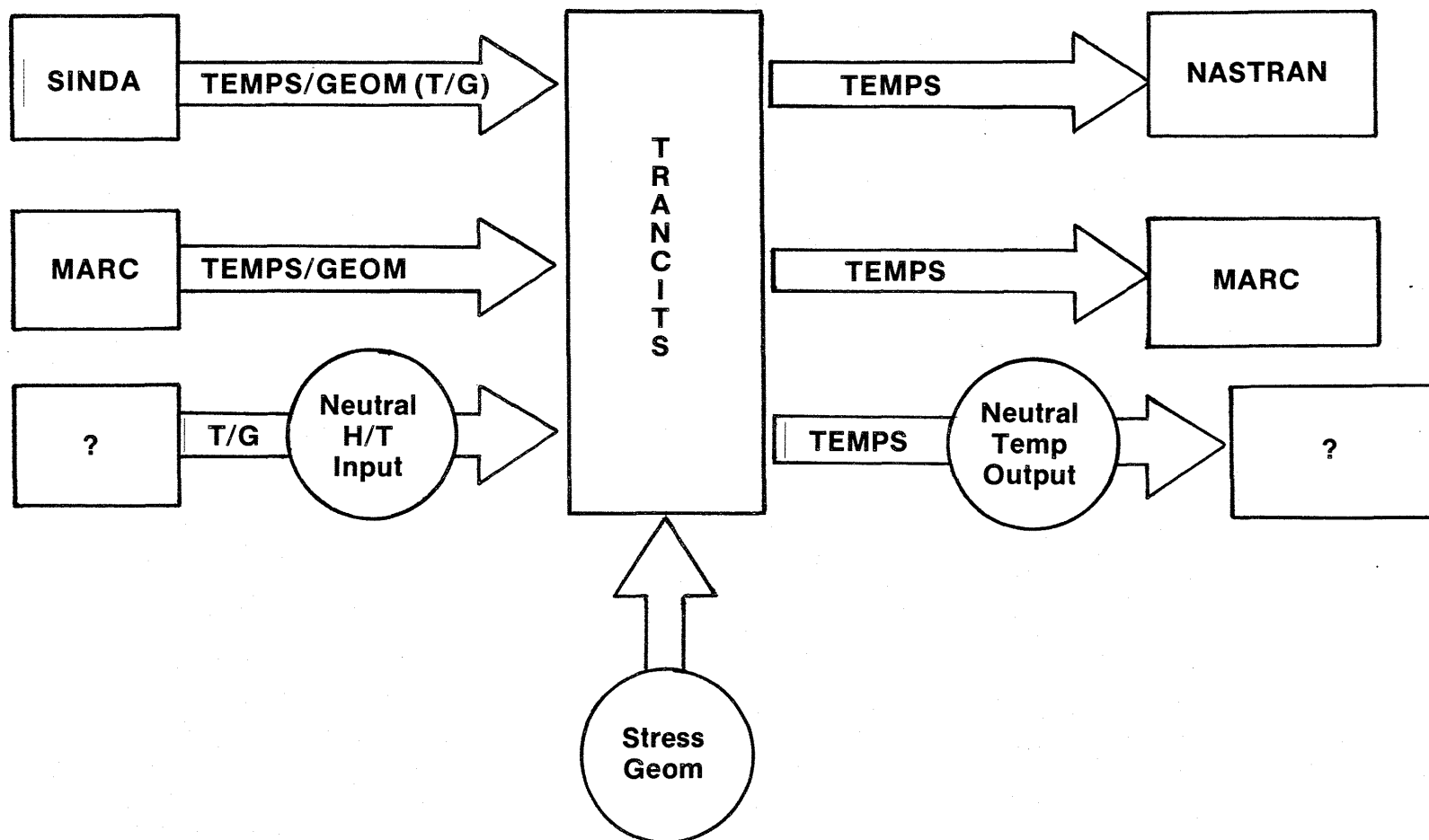


Figure 6-2. TRANCITS Schematic.

is not adequate for the transfer procedure. This file must be augmented by the SINDA geometry file. At first glance this seems to be a severe limitation of using TRANCITS in conjunction with any finite difference code. This is not the case because in practice large finite difference models are not created manually. Pre-processors automatically generate the volume and area data needed for the finite difference analysis. The input to these preprocessors is grid point coordinates and element connectivity. This input is exactly what is needed for the geometry portion of the TRANCITS input.

The input associated with the stress model is much simpler than that of the heat transfer model. To interface nodal temperature for the stress model all that is required is a file containing the name and rectangular cartesian coordinates of the stress points to be interfaced. This information is contained in the stress model coordinate file. If elemental centroid or elemental gauss point temperatures are requested, such as required by MARC, an additional file containing the stress element connectivity is also required.

The primary output of the transfer module is a neutral temperature output file. This file consists of the name of the stress node and its corresponding temperature for every transient time point requested. This file can easily be input into simple formatting routines and the stress point temperatures can be configured into any form required. In addition to this neutral output file, the program has the capability to automatically format the temperatures compatible with the NASTRAN or the MARC analysis codes.

6.3 FINITE DIFFERENCE CONSIDERATION

There are two primary differences between using a finite difference code versus a finite element code in conjunction with the transfer module. The first is the potential lack of geometry associated with the finite difference codes. This has been discussed in the previous section. The second is the elemental locations at which the thermal results are predicted. The basic output for a finite element heat transfer code is the temperature at the nodal points. This is ideally suited for the interfacing procedure, since these temperatures along with the assumptions associated with the thermal field inside the element can be used to accurately interpolate data within the element domain. On the other hand, finite difference codes produce temperatures at the element centroids and perhaps at the center of the element faces. Temperatures at these locations are much more difficult to use effectively and pose a problem. The approach taken in the transfer module is to convert these elemental temperatures into accurate temperatures at the elemental vertices. Once this has been accomplished, the interfacing technique proceeds in a manner similar to the one used for finite element heat transfer codes. This conversion of the elemental temperatures to nodal temperatures is performed in two steps. The first step is to map on an element by element basis the centroidal and face center temperatures to the element vertices. This mapping must be done in a fashion that is consistent with the assumptions governing the thermal distribution within the element. For a 3D linear finite difference volume, the assumption is made that the temperatures can vary linearly in all 3 directions

between the face centers and the centroid of the element. Given the temperatures at the 6 face centers and the centroid, the temperature, at the eight vertices of the element can be computed making use of the 3D tetrahedral isoparametric shape functions. The equations that must be solved are of the form:

$$T = [1, x, y, z] [c]^{-1} \bar{T}$$

where

T = temperature at vertice

x

y = coordinates of vertice

z

\bar{T} = known temperatures at the centroid and the three adjacent face centers.

where

$$[C]^{-1} = \begin{bmatrix} 1 & X_I & Y_I & Z_I \\ 1 & X_J & Y_J & Z_J \\ 1 & X_K & Y_K & Z_K \\ 1 & X_L & Y_L & Z_L \end{bmatrix}^{-1}$$

where

X_{IJKL}

Y_{IJKL} = coordinate of the tetrahedral

Z_{IJKL}

Note that $[C]^{-1}$ is only a function of geometry and therefore must only be computed once regardless of the number of transient time steps. This inverse is computed in closed form in the transfer module and is therefore efficient and accurate. Figure 6-3 illustrates the finite difference volume and the corresponding tetrahedral.

Note that during this initial stage the actual temperatures of the vertices are not directly computed, only the coefficients that relate the temperatures at the vertices to those at the known locations are computed.

The second step in this process is to compute a unique temperature at each vertice. This is done by weighting all of the vertice coefficients by a scale

- Face Centers
- ⊕ Element Centroid
- Corner Temperature Location

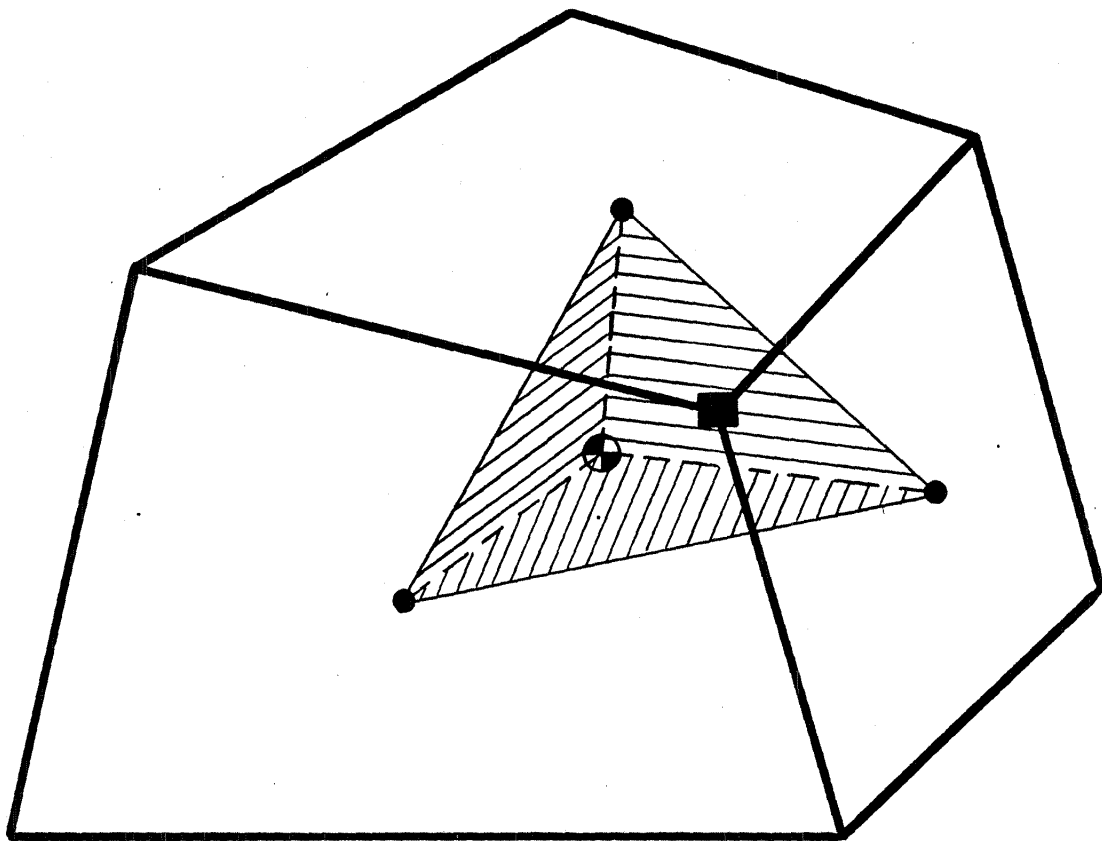


Figure 6-3. Finite Difference Volume and Corner Tetrahedral.

factor that is inversely proportional to the distance between the centroid of each element adjacent to the vertice and the vertice itself. This technique makes use of all of the surrounding temperature information to eventually produce a unique temperature at the vertice of each heat transfer element. Note that if the finite difference results are only at the centroid, the transfer module assumes the entire element was at that temperature. The weighting process however will generate different vertice temperatures since the temperature of the vertice depend on the geometry and the number of elements associated with that vertice.

6.4 3D SEARCH AND WEIGHTING COEFFICIENT ROUTINES

Two of the criteria that must be met by any transfer module are that the interfacing procedure be efficient and accurate. In order to satisfy the efficiency requirements, techniques were developed that rapidly determine which heat transfer element contains the stress points of interest. The approach taken was to invoke a "multi-step filtering" process. This process was developed and implemented such that most of the element candidates were eliminated using a very simple algorithm. The elements that passed through the initial "coarse" filter were then subjected to a more sophisticated test ("fine filter") that determined if the stress point was contained within the bounds of the element. This search technique, like the vertice coefficient routine is only a function of geometry and therefore must only be performed once regardless of the number of transient solutions interfaced. The simple coarse filter requires that the minimum and maximum values of the x, y, z coordinates be stored for each heat transfer element. This procedure is outlined below.

The stress point is outside of the heat transfer element if any of the following conditions are met:

$$X_s < X_{\min}$$

$$X_s > X_{\max}$$

$$Y_s < Y_{\min}$$

$$Y_s > Y_{\max}$$

$$Z_s < Z_{\min}$$

$$Z_s > Z_{\max}$$

where X_s , Y_s , Z_s are the coordinates of the stress point. For most models this technique drastically reduces the number of heat transfer elements that could possibly contain the stress point. Note that this procedure is valid for any order 3D heat transfer element as long as the maximum and minimum values are based on all of the nodes associated with the element.

The technique employed as the "fine filter" actually performs two tasks. First it determines if the stress point lies outside of the element, in which case the next element is considered. Second, if it is determined that the stress point is contained within the element, it automatically returns coefficients that relate the known temperatures of the vertices of the element to the desired temperature at the stress point location. This "fine filter" or weighting coefficient routine is based on the inversion of the isoparametric shape functions. The choice of these interpolating functions was based on many factors. These functions are widely used for many finite element heat transfer element formulations. They can be formed to represent a variety of temperature fields and therefore can be used to describe the variation within a finite difference volume. They are mathematically easy to manipulate and understand. The inversion process involves solving for the so-called local isoparametric coordinates (r, s, t) based on the global coordinates of the vertices of the element and coordinates of the stress point. Regardless of the order of the element the following equations can be written:

$$X = \sum_{i=1}^n \alpha_i X_i$$

$$Y = \sum_{i=1}^n \alpha_i Y_i$$

6.4.2

$$Z = \sum_{i=1}^n \alpha_i Z_i$$

$$T = \sum_{i=1}^n \alpha_i T_i$$

where X, Y, Z represents the geometry

T represents the normal field

X_i, Y_i, Z_i are the coordinates of the vertices of the heat transfer element

N is the number of vertices

α_i are the isoparametric shape functions

The problem is to solve for local coordinates (r, s, t) that correspond to the global coordinates of the stress point (X_p, Y_p, Z_p). The relationships between

the shape functions and the local coordinates are well known once the order of the element has been fixed. Substitution of these relations into the above summation along with the equating of X to X_p , Y to Y_p and Z to Z_p yields a set of non-linear equations with r , s and t as unknowns. The local coordinates become functions of each other, global vertice coordinates and the coordinates of the stress point.

That is:

$$r_p = f(r_p, s_p, t_p, X_i, Y_i, Z_i, X_p, Y_p, Z_p)$$

$$s_p = f(r_p, s_p, t_p, X_i, Y_i, Z_i, X_p, Y_p, Z_p) \quad 6.4.3$$

$$t_p = f(r_p, s_p, t_p, X_i, Y_i, Z_i, X_p, Y_p, Z_p)$$

These equations can now be solved numerically to yield values of r_p , s_p , t_p . Once the values of the local coordinates are known, a simple test determines if the stress point is contained in the heat transfer element. The stress point is not contained in the element if any of the following conditions are met.

$$-1 > r_p > 1$$

$$-1 > s_p > 1$$

$$-1 > t_p > 1$$

If the point is found to be contained with local coordinates can be substituted into the expressions for the shape functions and the thermal weighting coefficient can then be computed.

This is an ideal method for this interfacing procedure, it finds the correct heat transfer element and at the same time produces accurate weighting coefficients. Figure 6-4 illustrates this mapping technique.

For this study the 3D linear isoparametric formulation was chosen as the interpolating functions for the heat transfer elements. The set of non-linear equations described in Equation 6.4.3 were solved by using a fixed point iteration technique in conjunction with an "educated initial value" procedure. These initial values are computed as:

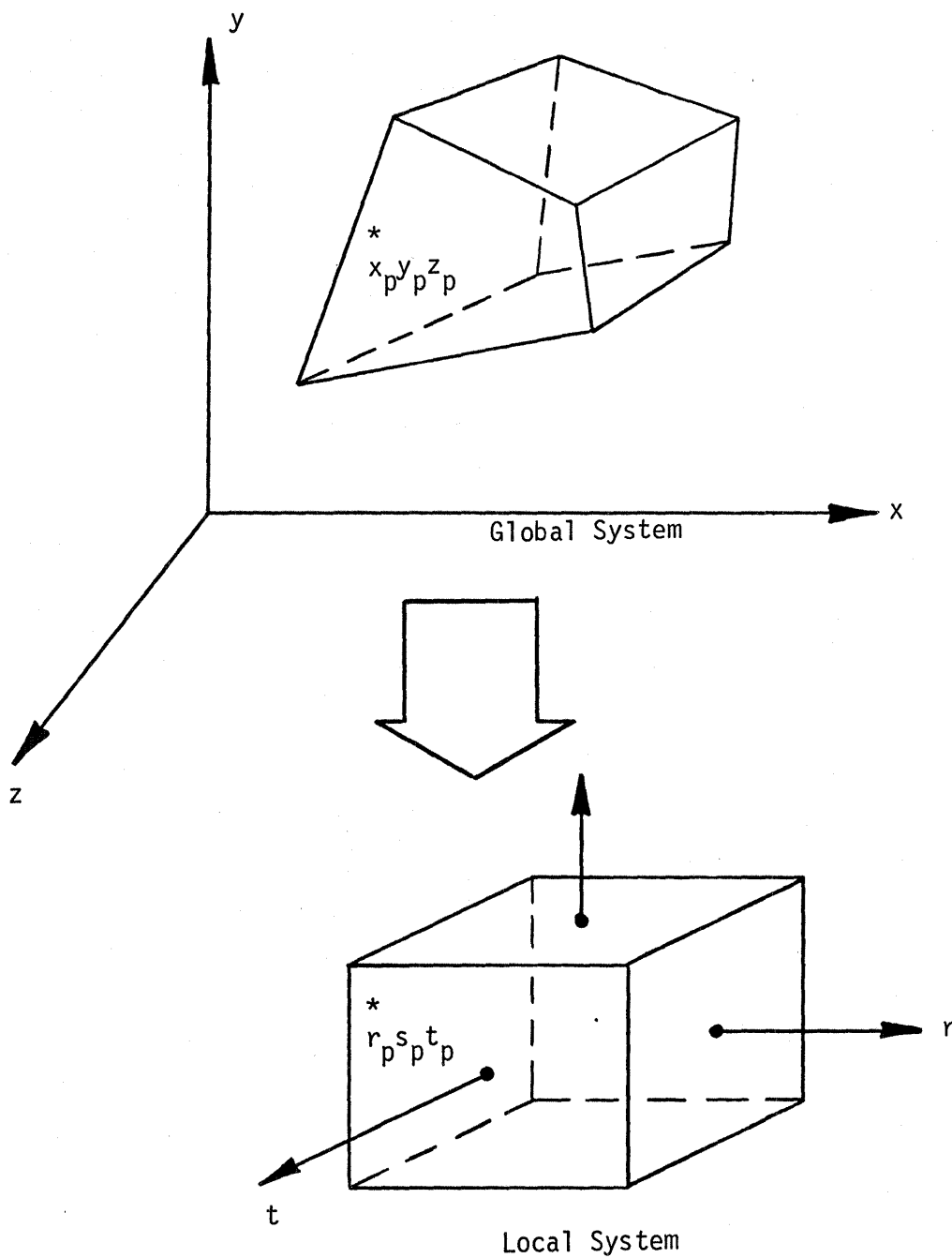


Figure 6-4. Isoparametric Mapping of Global Coordinates To Local Coordinates.

$$r_{pi} = \frac{X_p - X_{min}}{X_{max} - X_{min}}$$

$$s_{pi} = \frac{Y_p - Y_{min}}{Y_{max} - Y_{min}}$$

6.4.5

$$t_{pi} = \frac{Z_p - Z_{min}}{Z_{max} - Z_{min}}$$

These initial values correspond to the correct values of r_p , s_p , t_p if the heat transfer element is a rectangular parallel piped. The initial^pguesses^p are used in conjunction with the righthand side of the non-linear equation to produce new values of the local coordinates. This iteration continues until convergence is achieved. The initial implementation of this technique showed it to be very efficient and fairly insensitive to element distortion. Several highly distorted 8-node 3D elements were tested. Figure 6-5 shows two of these elements. Convergence was achieved for all points attempted using these elements. Once this technique was installed into the transfer module and actual problems were interfaced, it was discovered that although the iterative procedure was not sensitive to the shape of the element, it was very sensitive to the orientation of the element with respect to the global axis system. It was found that for some configurations, the technique became unstable and either diverged or converged to an undesirable root of the equations.

Logic was added to the transfer module to avoid these stability problems. First, a coordinate transformation was applied to the heat transfer element to align the direction defined by I and J joints of the element with the global X axis. The iterative technique is then used to solve for the local coordinates of the stress point. If it fails to converge to the correct root, the program reorients the element such that a new vertex of the element is now the I joint. The alignment transformation is then performed and the iterative solution is repeated. If it still fails to converge a new vertex is chosen for the I joint and the process is again repeated. If convergence is not achieved after all the vertices have been used, the stress point is not contained in this heat transfer element.

A 400 element airfoil model, shown in Figure 6-6 was run using this version of the transfer module. The stress points used were the centroids of the elements and 8 additional points per element that were very near the corners of the element. There were a total of 3735 stress points. After the first transformation 3728 of the stress points converged. The other 7 stress points converged after the additional reorientation of the element was completed.

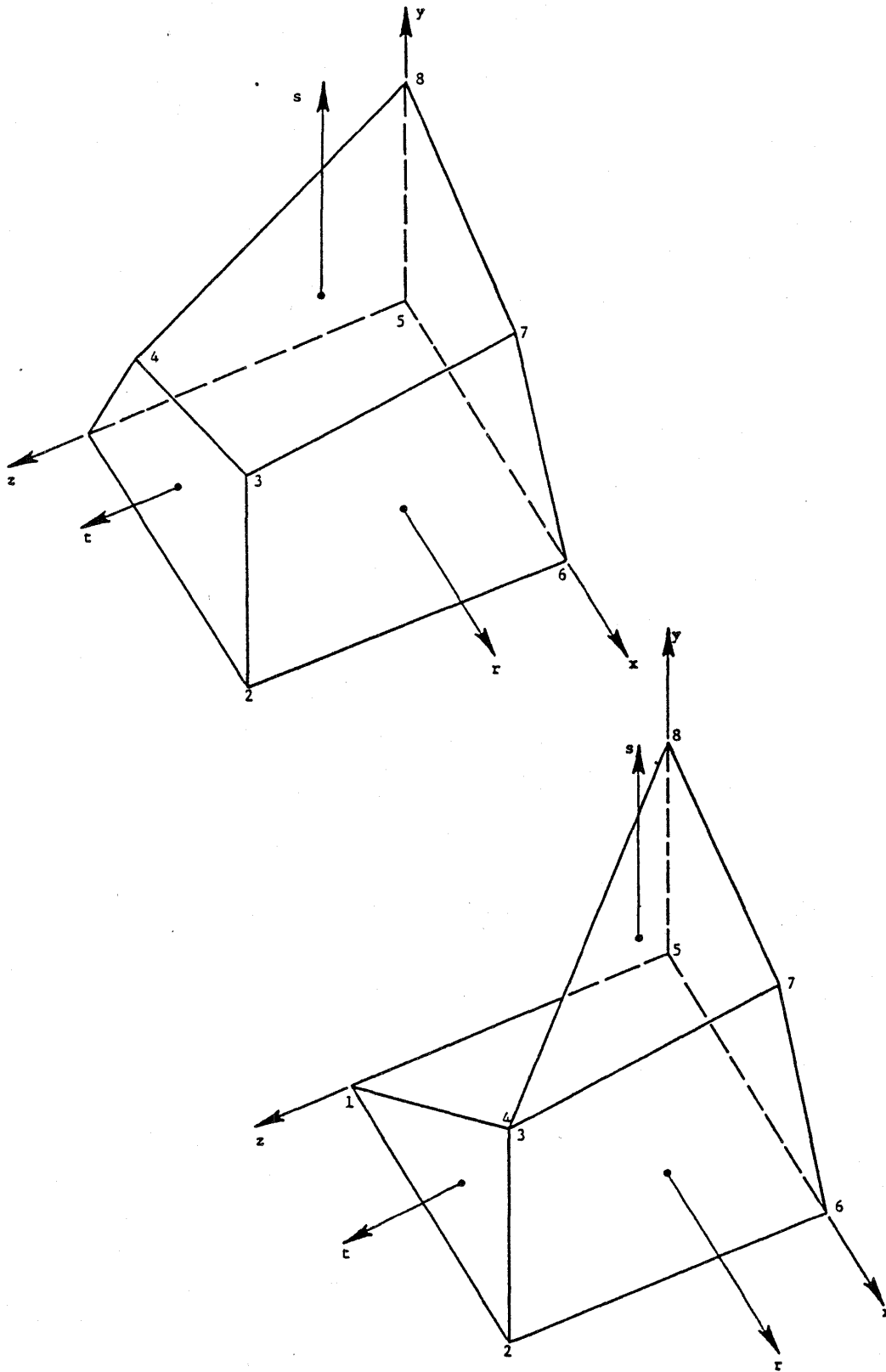


Figure 6-5. Highly Distorted Elements Tested.

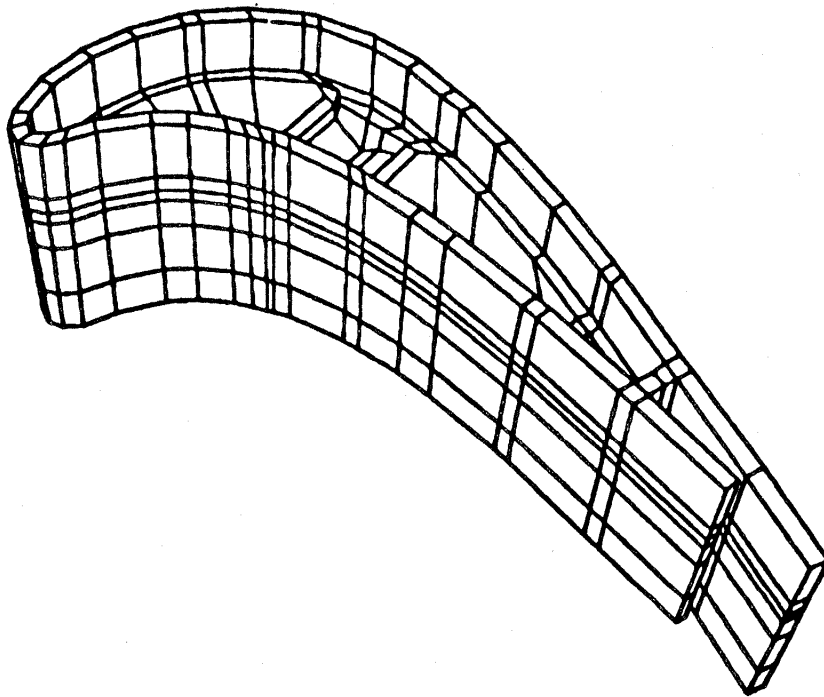


Figure 6-6. 400-3D Element Air Foil Model Used As Test Case.

These results show that the initial transformation is adequate for most of the stress points and that the additional computation required for the vertex reorientation is only necessary for a very small percentage of stress points.

As stated earlier the existing transfer module inverts the 3D linear isoparametric shape function. The technique used, however, can be applied to any order element with only small impact on the coding of the transfer module.

6.5 EXTERIOR POINT SURFACING ROUTINE

In order for the transfer module to be a useful tool, it must deal with stress points that lie slightly outside of the heat transfer module. This situation occurs for many reasons in practice, the most common being part tolerances used to create the heat transfer model and the stress model. It is not the intention of the transfer module to produce temperatures for stress points that are in areas outside of the domain of the thermal analysis and therefore no extrapolation of temperatures are performed. On the other hand, if a stress point exists that physically represents the surface of the part and that point because of slight tolerance mismatches is not contained in the heat transfer element on the surface, the transfer module should produce a temperature that is representative of the surface temperatures. This has been accomplished in TRANCITS by implementing an exterior surfacing routine. This technique also makes use of the local coordinates (r, s, t). If the stress point lies outside the element, one of the local coordinates will be greater than one. For example,

```
r = .6  
s = -1.2  
t = -.8
```

This point is not contained in a heat transfer element since

```
s    1
```

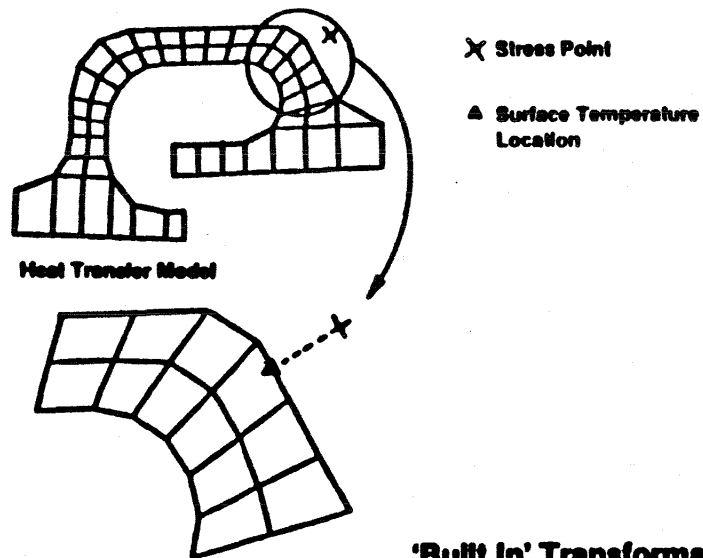
A point that lies on the surface near the stress point would have local coordinate values of

```
r = .6  
s = -1  
t = .8
```

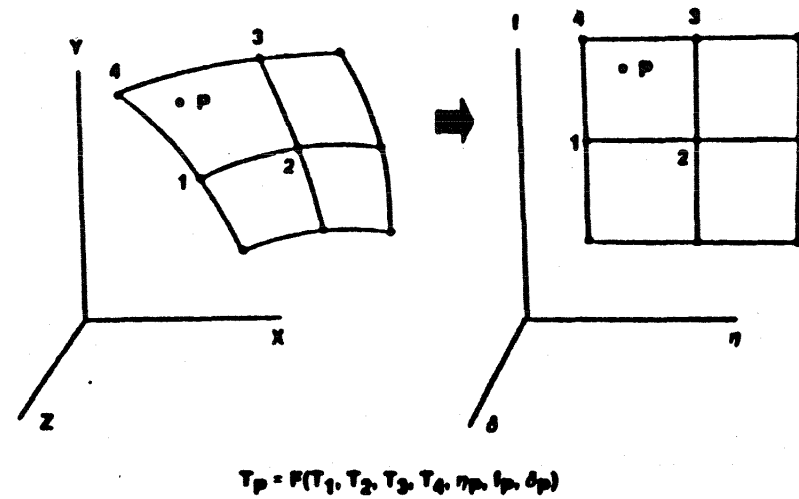
The distance between (r, s, t) and (r , s , t) can easily be computed in physical space and stored. The heat transfer element with the smallest distance can then be used as the closest element and the temperature of the surface point can be produced. This allows the transfer module to predict an approximate temperature for the stress points that lie outside the heat transfer model. The program flags all stress points that exhibit this behavior and also prints out the distance between the actual coordinates of the stress

point and the point on the surface for which the temperatures were computed. Users should pay close attention to these points and judge if the distance outside the model is acceptable for the analysis being performed. Figure 6-7 illustrates this surfacing technique along with the options of built in transformation and isoparametric mapping techniques.

Exterior Point Method



Isoparametric Mapping



'Built In' Transformations/Windowing

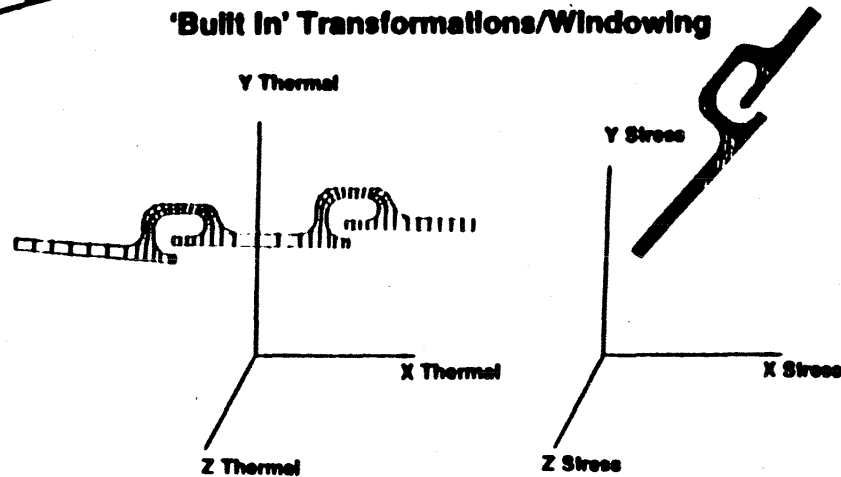


Figure 6-7. Enhanced Program Features.

7.0 VERIFICATION AND DOCUMENTATION OF THE TRANSFER MODULE

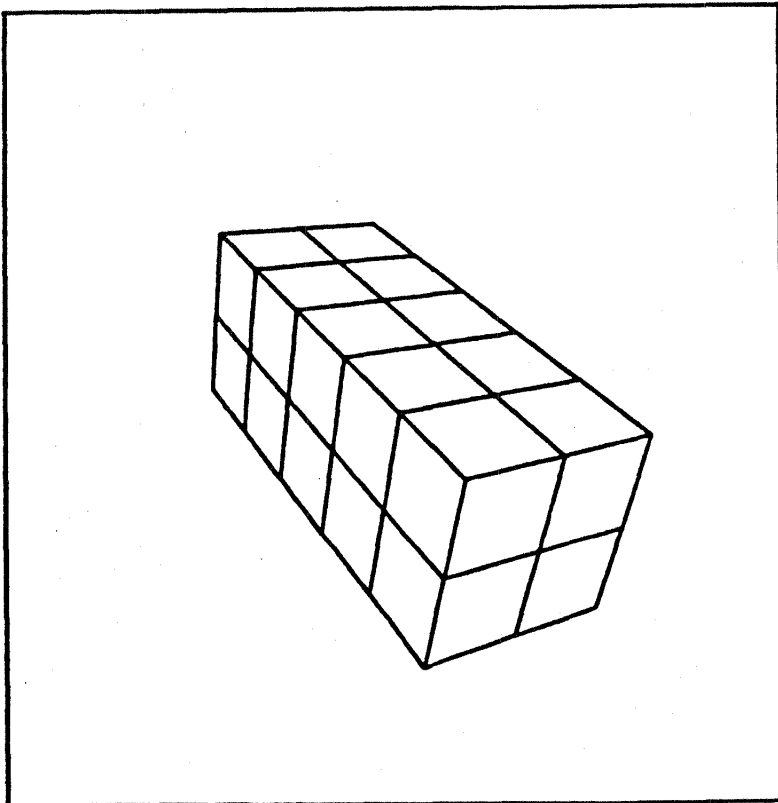
Two objectives were met by the verification and documentation phase of this contract. The verification subtask provided the necessary test cases to ensure that all of the options implemented were producing accurate and efficient results. Indeed many efficiency improvements were identified and implemented in this phase. The purpose of the documentation subtasks was to provide information required to use the transfer module as well as the documents needed to understand and possibly alter the coding for specific problems.

7.1 VERIFICATION OF THE TRANSFER MODULE

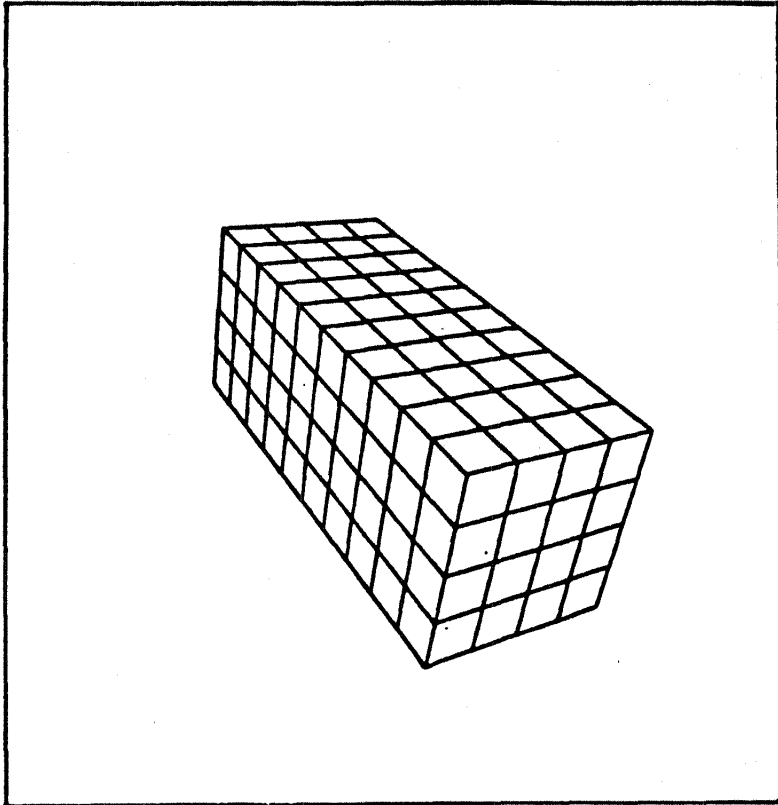
Verification of the transfer code was performed in parallel with the development effort. Each of the modules in the code were tested to ensure that the input and output of those modules were consistent with all file structures. All user options were executed and thermal results from both SINDA and MARC were directly input and processed by the transfer code. Nodal temperatures for NASTRAN were produced as well as elemental centroid temperatures for MARC.

Formal verification of the mapped thermal results were conducted using three different 3D models. The primary objective of these test cases was to ensure that the thermal data produced by the heat transfer codes was not degraded in the mapping to the stress analysis models. The verification procedure involved comparing the results of the heat transfer code to the interpolated temperatures produced by TRANCITS. Comparisons were made using two different methods. The first method plotted the heat transfer results and the TRANCITS results against the model geometry for various regions of the 3D models. These plots were used to verify that the thermal gradients predicted by the heat transfer analysis were indeed mapped into the stress models. The second comparison method utilized the PATRAN Reference 3 postprocessing program. Colored isotherms were created for the stress model using the interpolated temperatures and then for the heat transfer model using the original heat transfer results. These two sets of isotherms were then compared for a qualitative check on the thermal gradients.

The first test case was a simple model of a rectangular prism. The heat transfer mesh was composed of a 2x2x5 grid, the stress mesh was much finer and had a mesh density of 4x4x10. These models are shown in Figure 7-1. The boundary conditions applied to the heat transfer model give rise to a linear thermal gradient along the length of the prism. The transfer module was used to map these temperatures into the finer stress model and Figure 7-2 shows a plot of the heat transfer code temperatures and the TRANCITS temperatures versus the distance along the prism. The figure shows that for this simple case the interpolated values agree exactly with those predicted by the heat transfer code. Figure 7-3 and 7-4 show the isotherms for the heat transfer model and stress model respectively. Visual comparison of these isotherms again indicate the same thermal gradients in both models.



HEAT TRANSFER MODEL



STRESS MODEL

Figure 7-1. Prism Test Case.

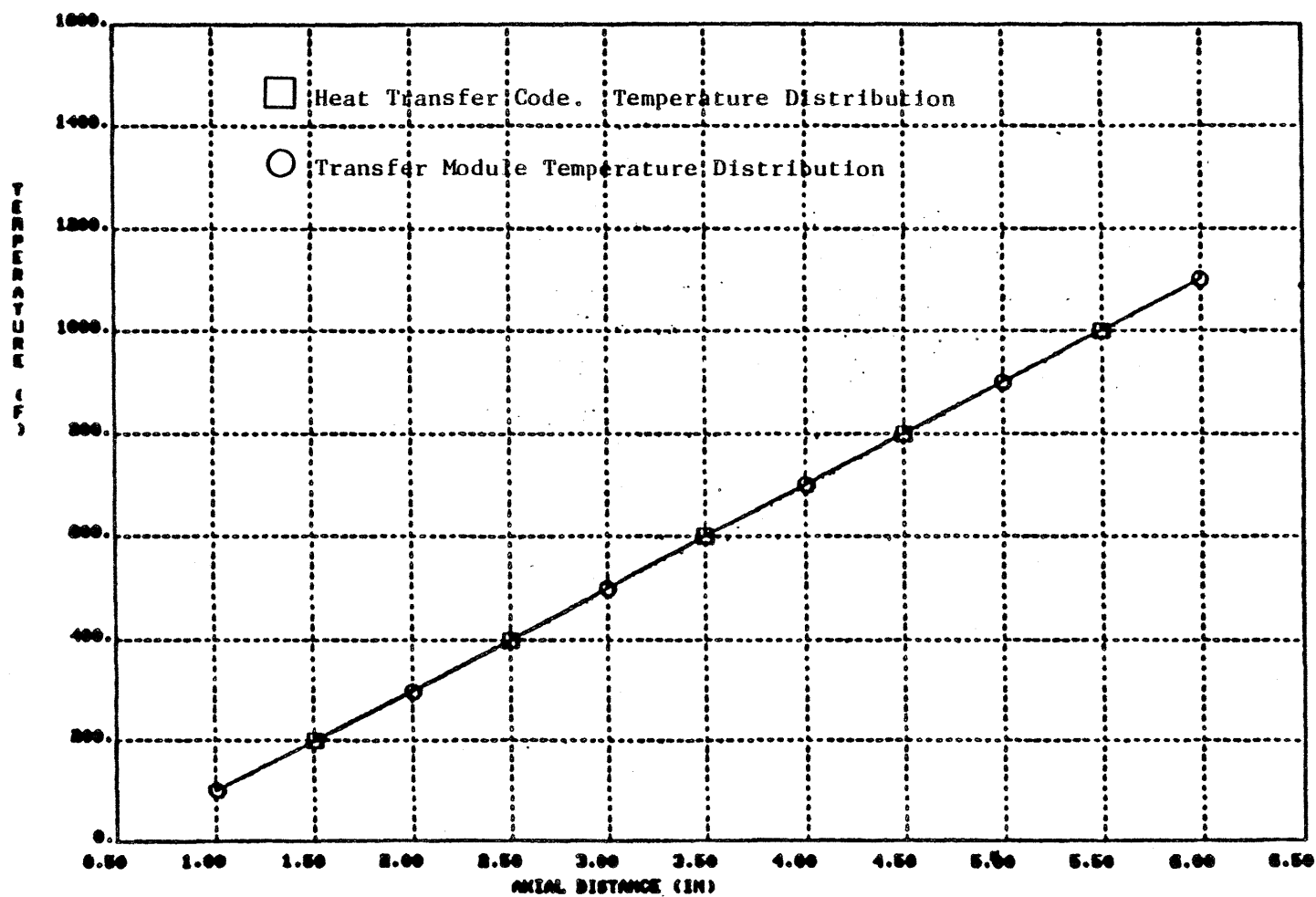


Figure 7-2. Temperature Versus Axial Distance For Beam Test Case.

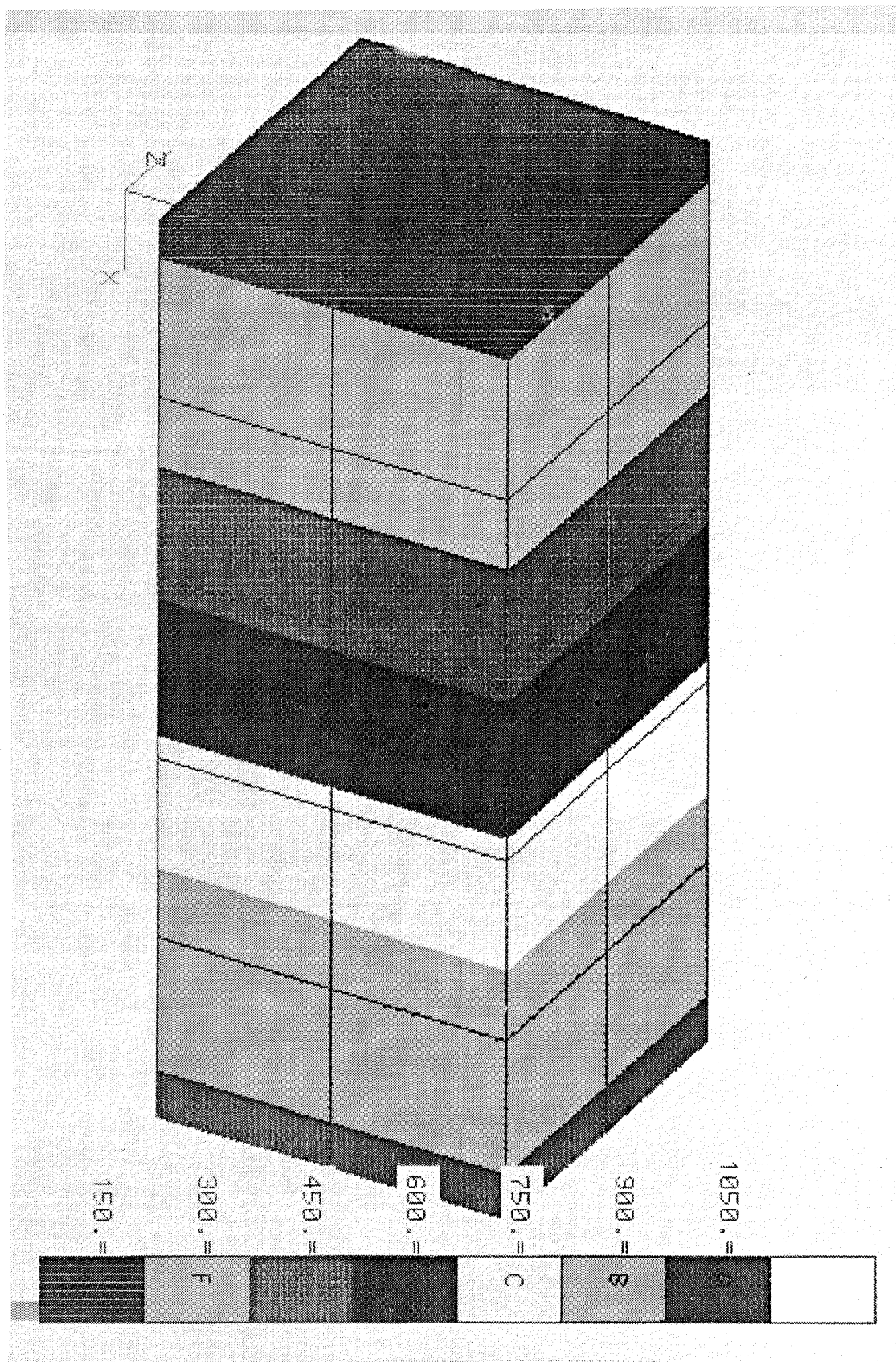


Figure 7-3. Heat Transfer Model Isotherm.

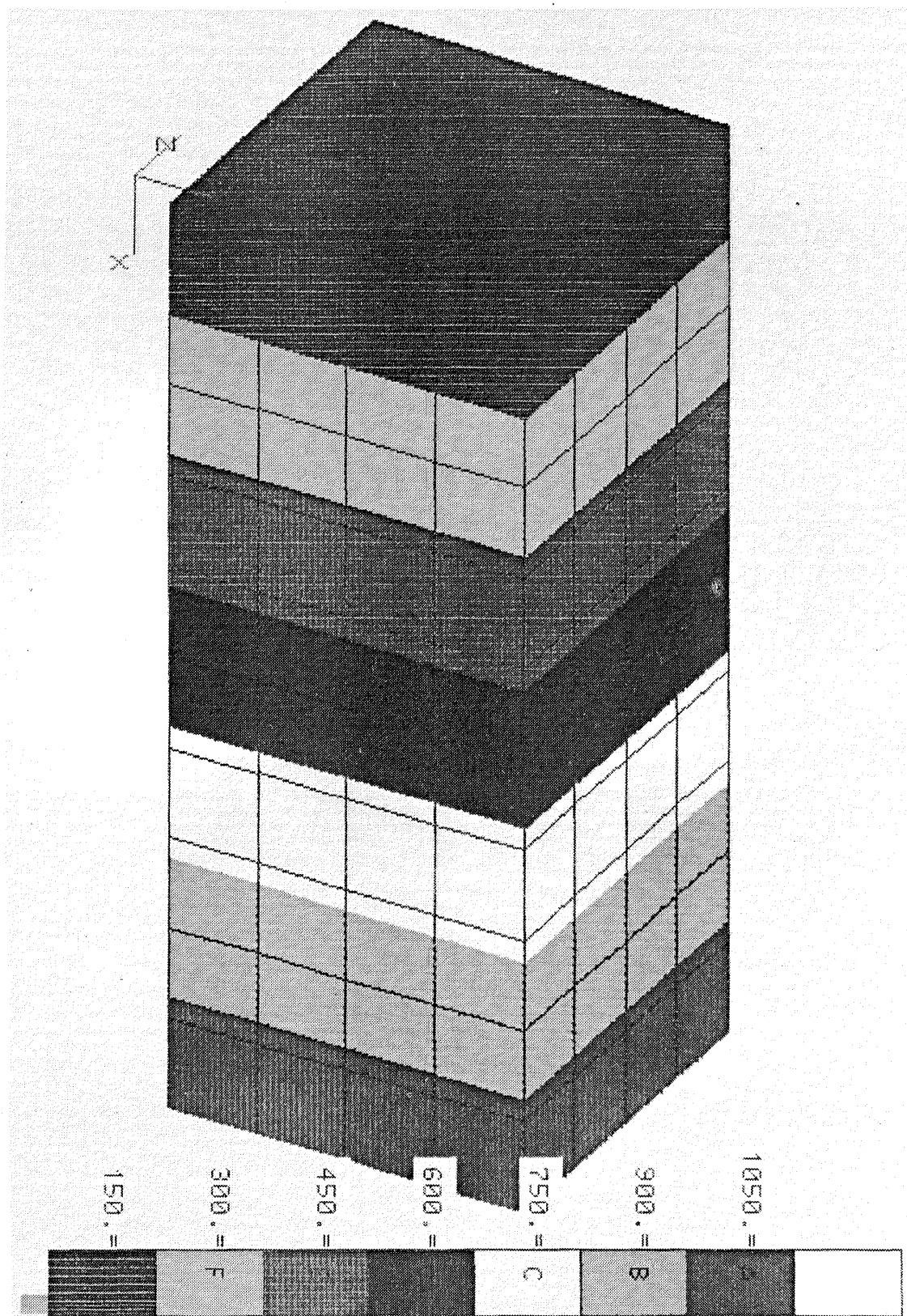


Figure 7-4. Stress Model Isotherm

The model used for the second test case is shown in Figure 7-5. It is finite difference model of the tip of a turbine blade composed of approximately 450 heat transfer elements. For this case the stress model mesh density was identical to the heat transfer model. The transfer module was used to map the heat transfer temperature at the centroid of the elements and the face centers of the element sides to the vertices of the element. The boundary condition applied to the heat transfer model represent realistic conditions for a turbine blade and give rise to gradients in all directions. Figures 7-6 thru 7-11 show plots of the original heat transfer temperatures and the transferred corner temperatures against a local distance parameter. These plots are for 3 spanwise sections through the airfoil (Tip, Middle, Root Sections) for both the concave and convex sides of the blade. The local distance parameter is measured from the leading edge on the concave side and the trailing edge on the convex side. The first two curves on each plot represent the temperatures for a layer of heat transfer elements just above and just below the corner node temperatures. As the plots show, the corner temperature distribution not only has the same trends as the original heat transfer temperature distribution but the value of the temperature at each corner node falls in between the temperature above and below the corner as computed by the heat transfer code.

These comparisons provide excellent verification of the accuracy of the transferring technique.

Models representing the final test case are shown in Figure 7-12. These models represent a 3D sector of a combustor liner. The mesh density for the heat transfer model is finer than the stress model in some locations and coarser in others. The boundary conditions used in the heat transfer model once again represent typical engine conditions. Figures 7-13 and 7-14 show isotherms for both heat transfer and stress grid respectively. These isotherms also verify that the gradients are essentially the same for both models.

7.2 DOCUMENTATION OF THE TRANSFER

The documentation of this code was divided into two segments. The first was the TRANCITS User's Manual. This manual describes in detail all of the input and output file contents for the options implemented in this version of the transfer module. It shows examples of these files and discusses how the code should be executed. It also has a complete description of all the user input variables and what defaults if any are used if these inputs are omitted. The manual specifies the current limitations of the code in regard to maximum size of problems that can be interfaced. The manual should provide all of the necessary information required to use the transfer modules.

The second involved the documentation of the coding of the transfer module. Listings and prologs of each subroutine used and a flow diagram of the entire code have been included in this report in Appendices C and D. These prologs are part of the source code and they give a brief description of the purpose of the subroutine. They also describe all of the calling arguments and whether these arguments are inputs or outputs to the routine. All files used in the routines are specified as well as any routines that are called

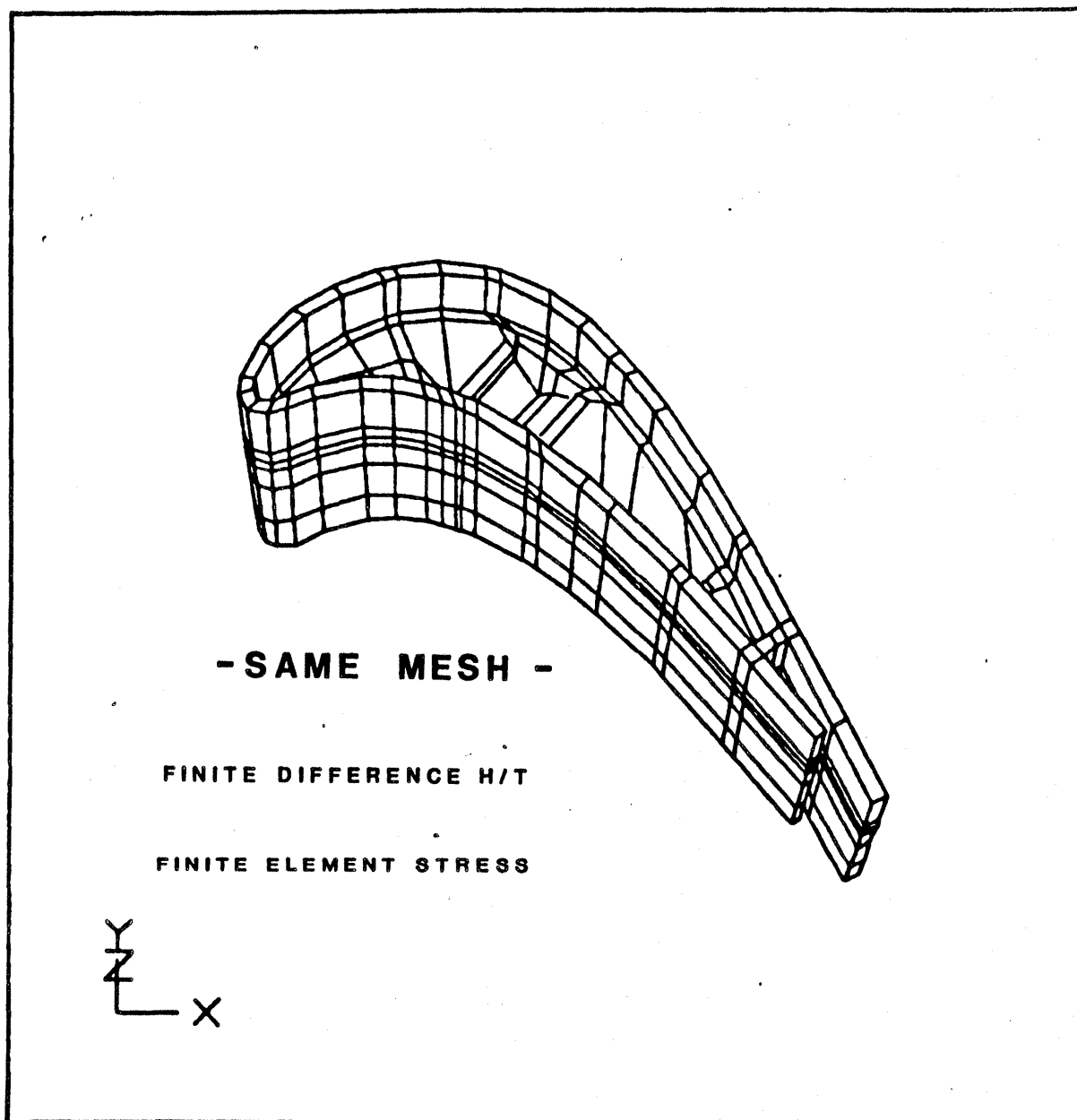


Figure 7-5. Airfoil Test Case Model.

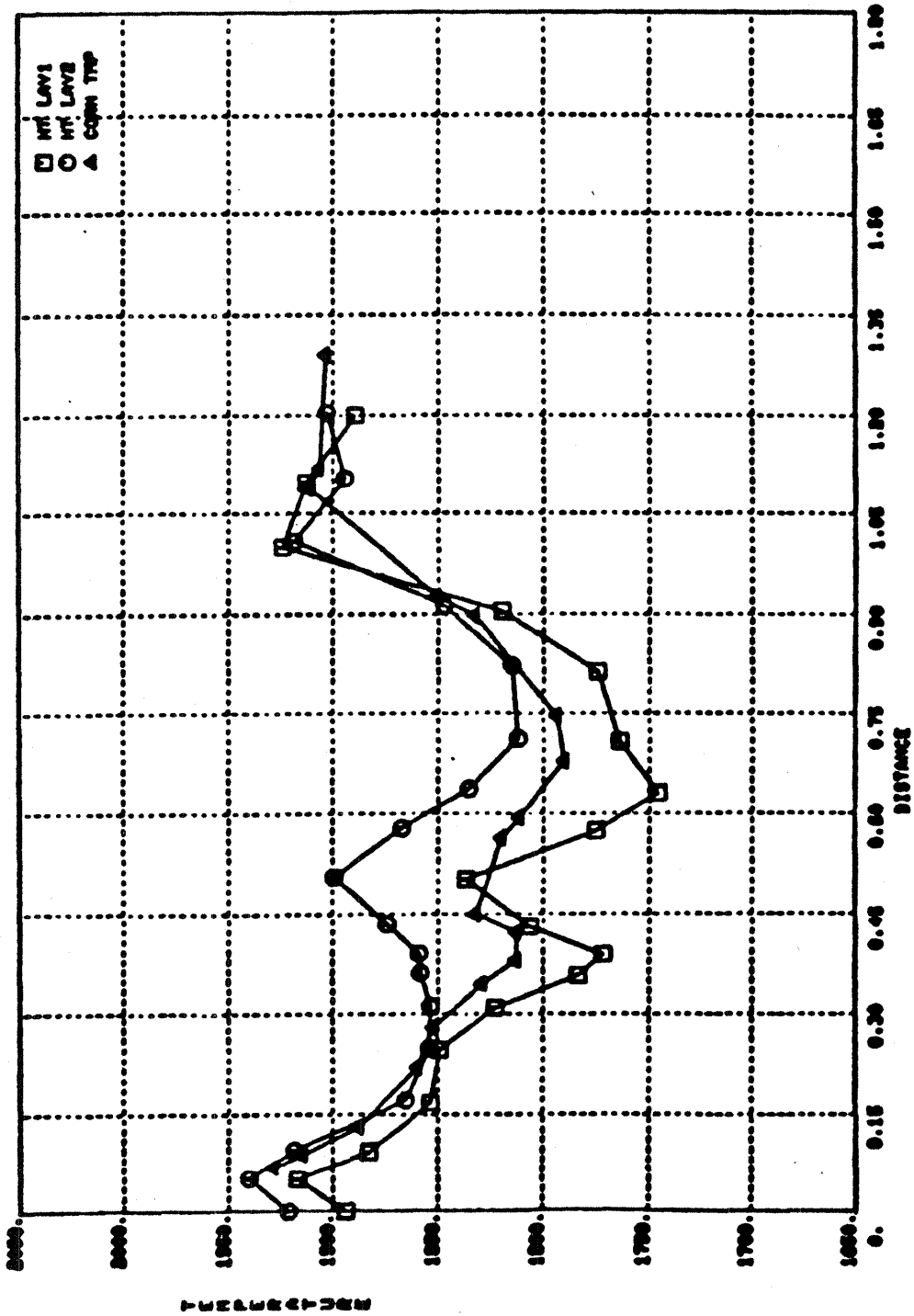


Figure 7-6. Temperature Verification Plots Concave Side Tip Section
Large Model (Airfoil).

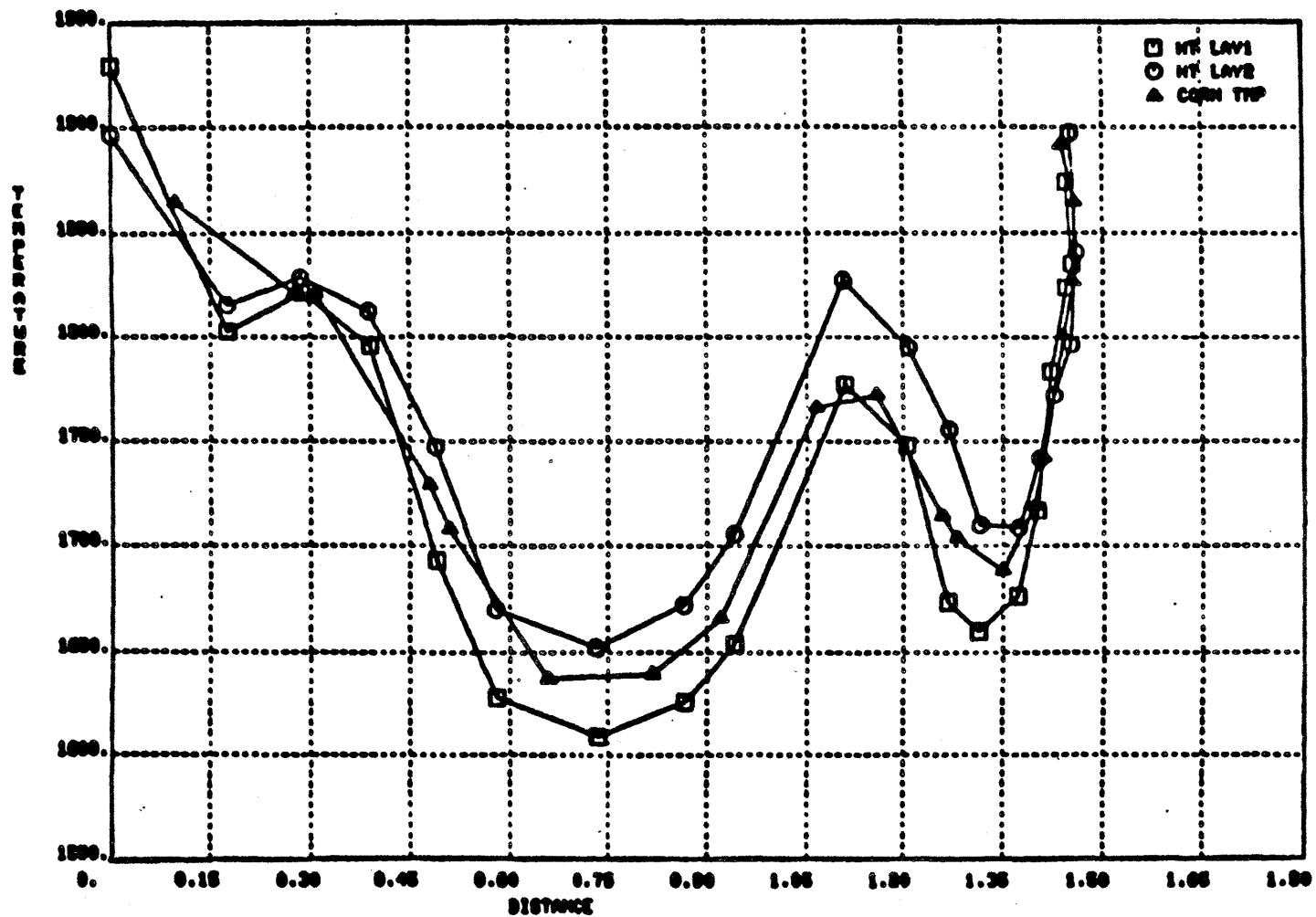


Figure 7-7. Temperature Verification Plots Convex Side Tip Section
Large Model (Airfoil).

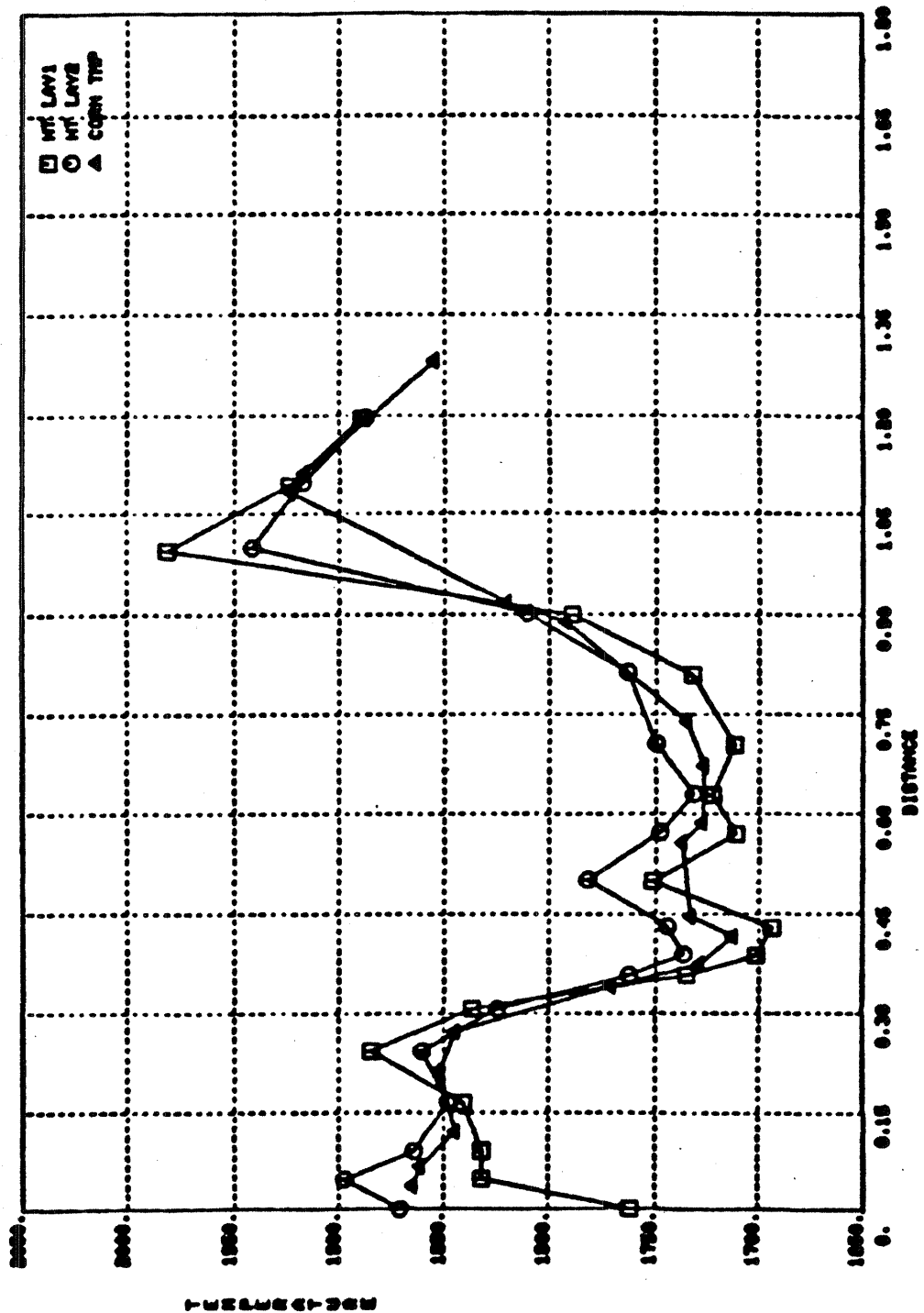


Figure 7-8. Temperature Verification Plots Concave Side Mid Section Large Model (Airfoil).

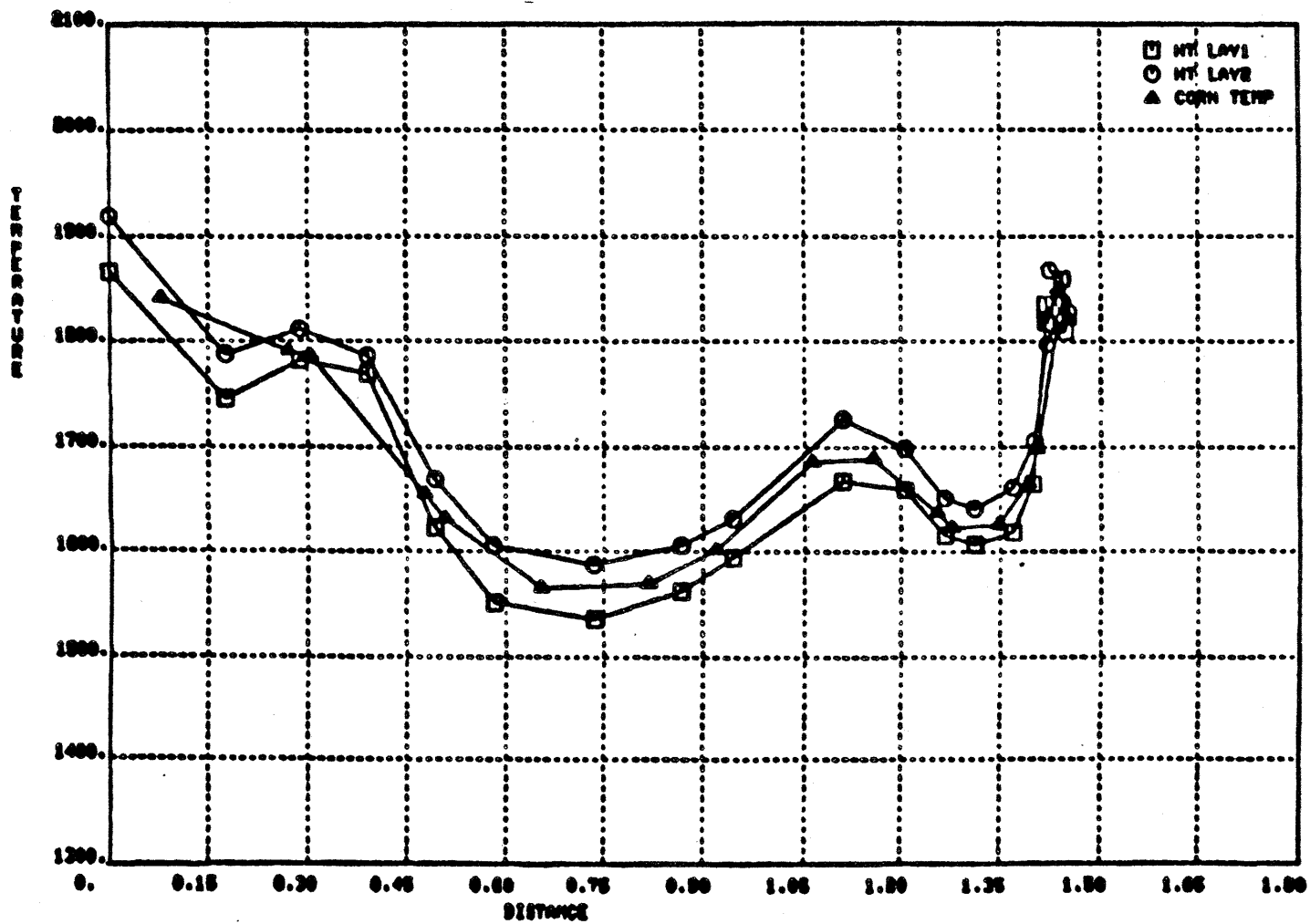


Figure 7-9. Temperature Verification Plots Convex Side Mid Section Large Model (Airfoil).

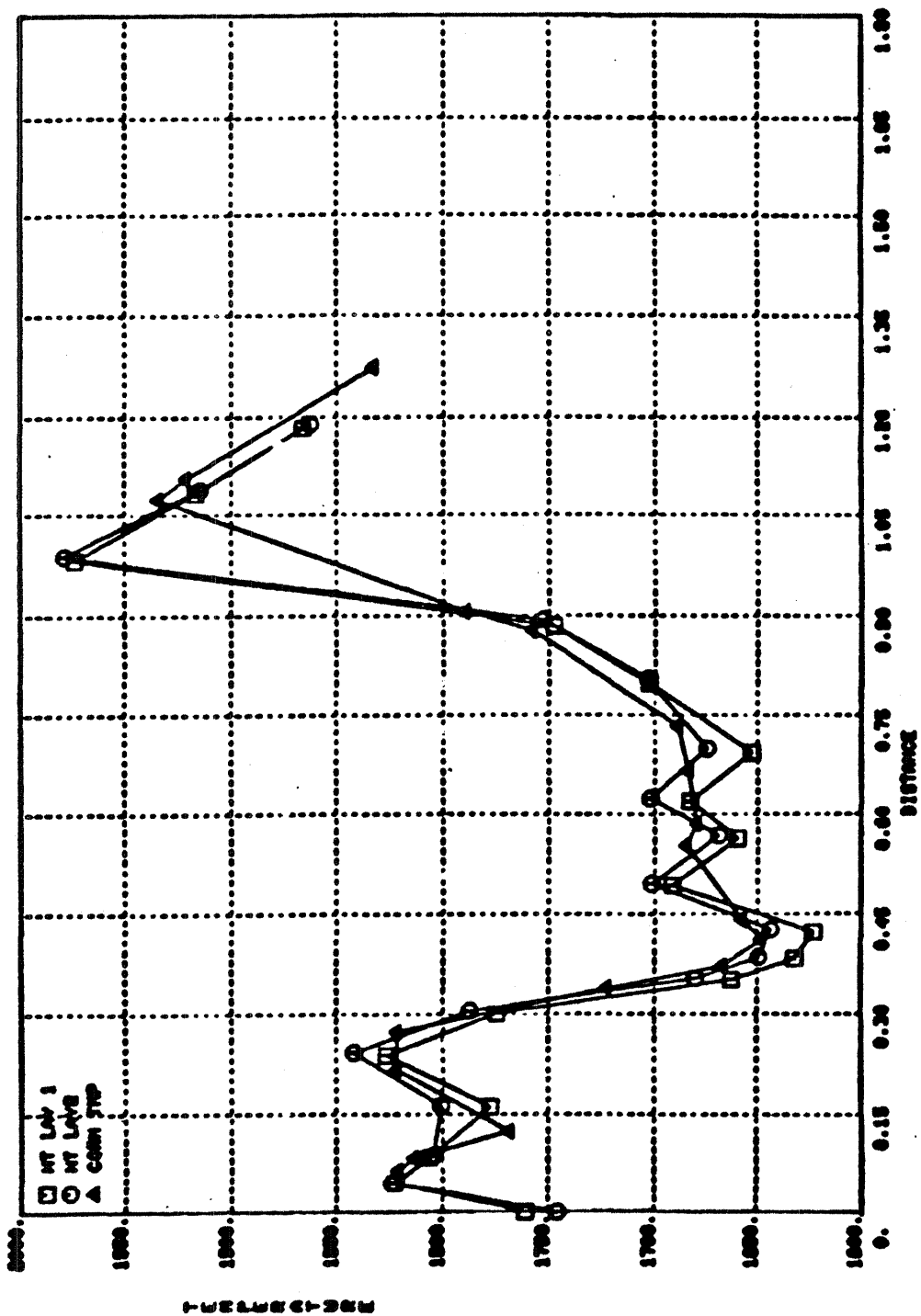


Figure 7-10. Temperature Verification Plots Concave Side Root Section Large Model (Airfoil).

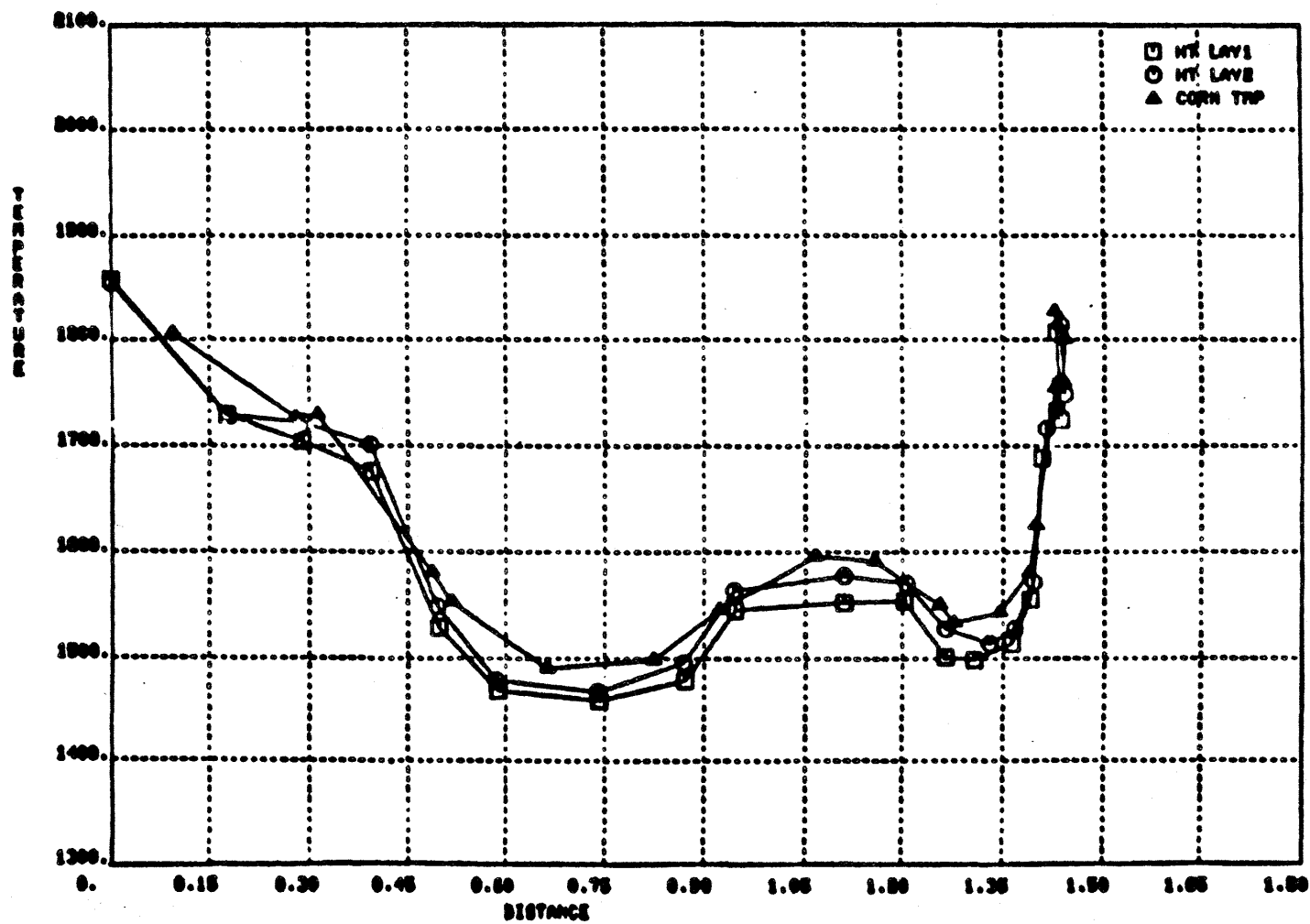
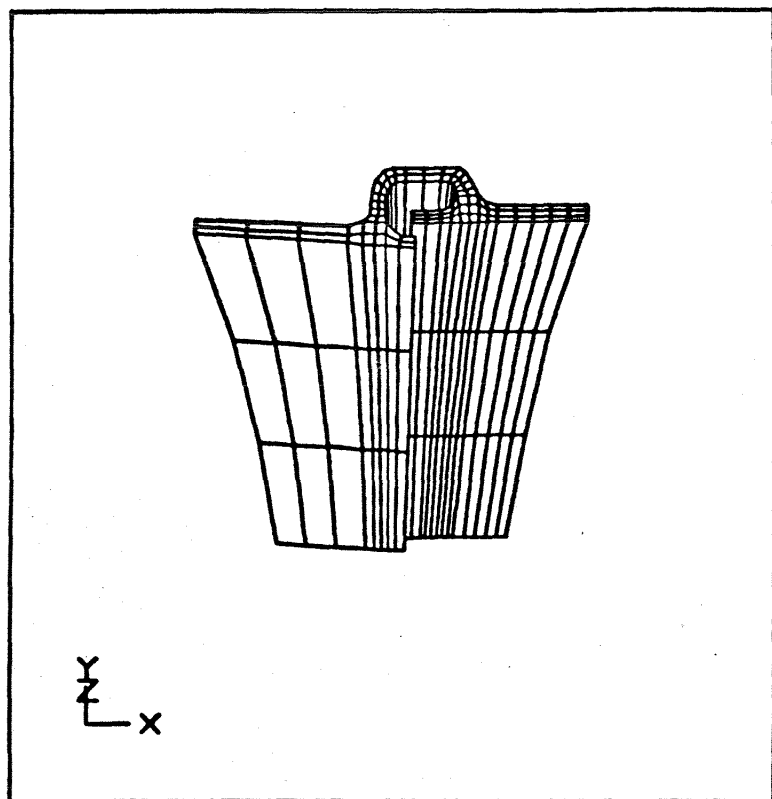
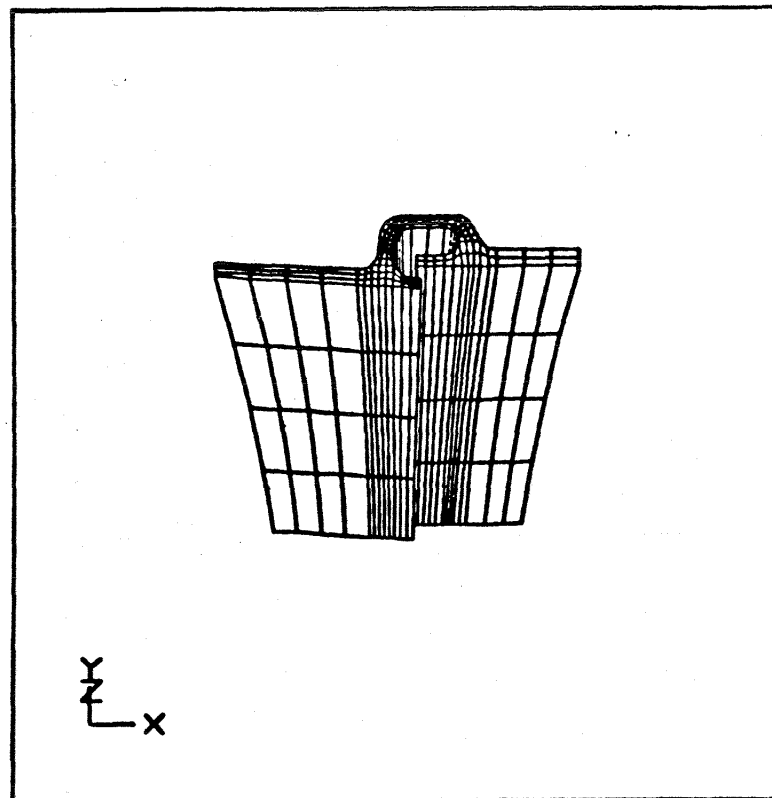


Figure 7-11. Temperature Verification Plots Convex Side Root Section Large Model (Airfoil).



HEAT TRANSFER MODEL



STRESS MODEL

Figure 7-12. Combustor Liner Test Case Models.

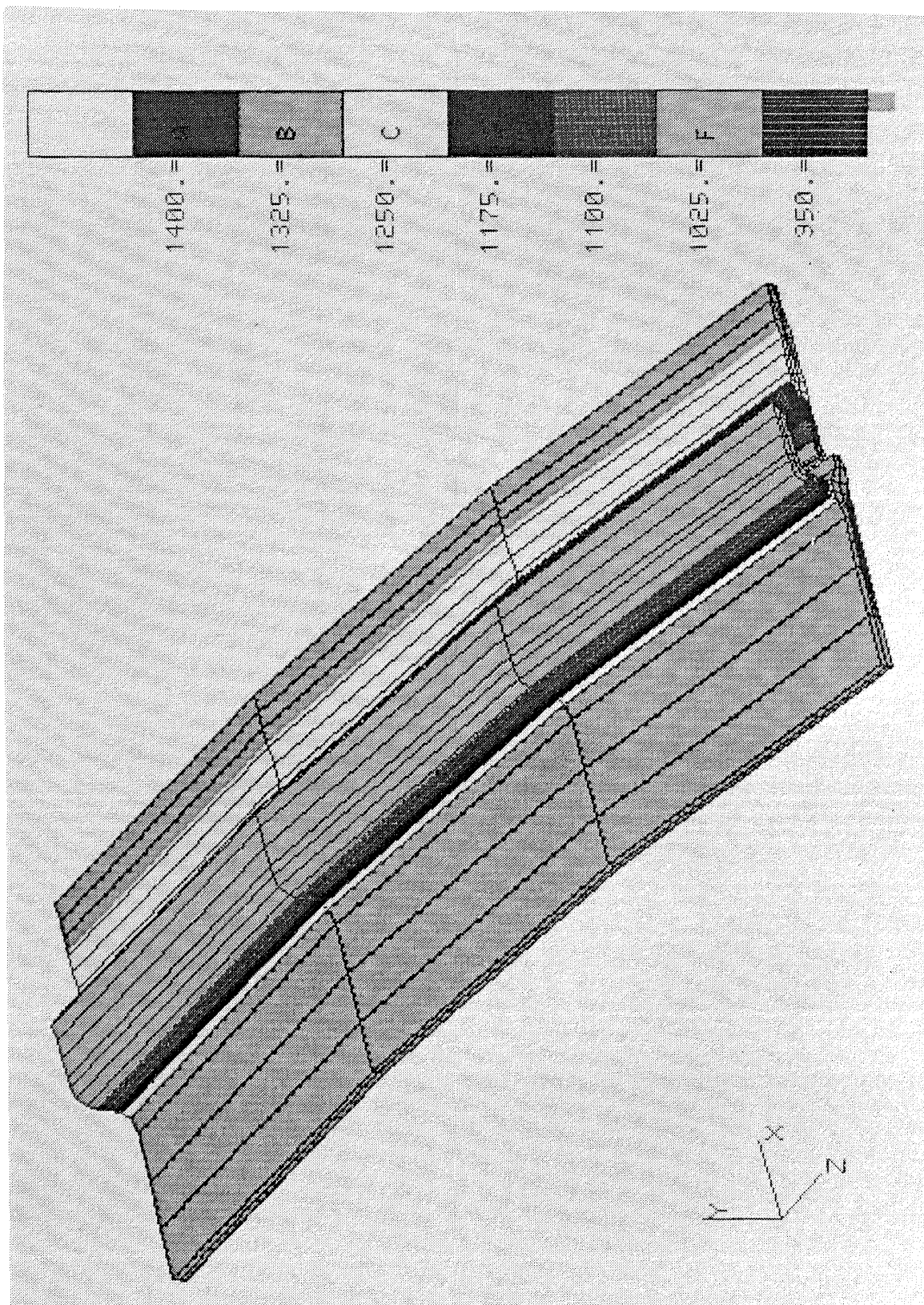


Figure 7-13. Combuster Liner Heat Transfer Model Isotherms.

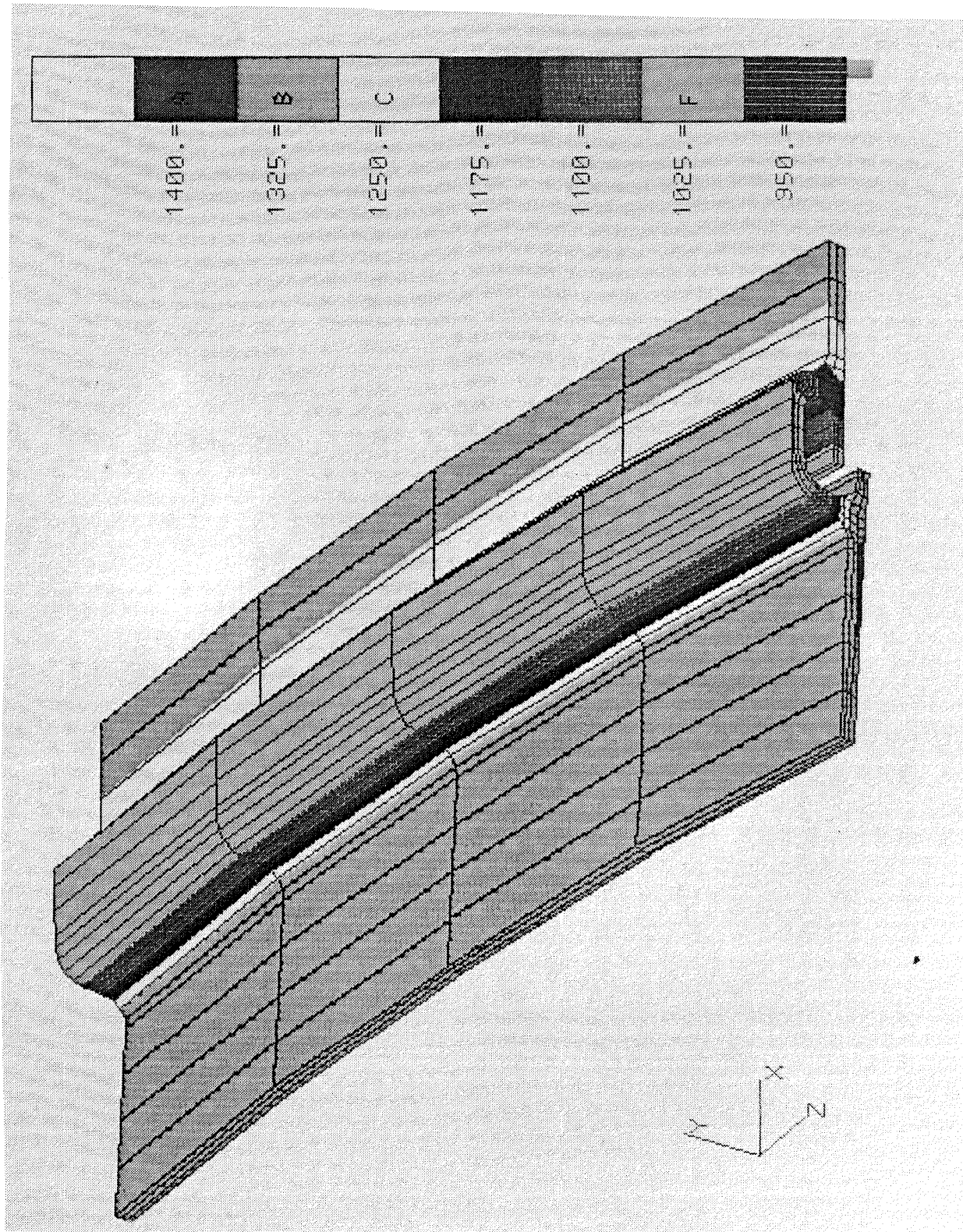


Figure 7-14. Combuster Liner Stress Model Isotherms.

from the subroutine. Commons that are used in the routine are also noted. If there are any special limitations or comments, these are described in the prolog.

The flow diagram gives a concise picture of the program flow and provides a convenient reference for which subroutines are a part of each module. This and the program documentation should enable any user to understand and perhaps adapt this transfer module to their individual needs.

8.0 CONCLUSIONS

The TRANCITS software package accurately and efficiently transfers thermal data from a 3D heat transfer study to a 3D stress analysis mesh of different elemental density. The heat transfer analysis can be performed using finite element or finite difference techniques. TRANCITS currently supports the output of the MARC and SINDA heat transfer codes directly and will format the thermal data compatible with the input requirements of NASTRAN and MARC. In addition, the neutral heat transfer input file and neutral temperature output file can be used to interface the results of any heat transfer code to any structural code. The transfer module has been written in a modular fashion and is easy to implement and modify. The technology developed to transfer thermal data forms an excellent foundation for the transfer of other engineering parameters such as pressures, loads and displacements.

This Page Intentionally Left Blank

APPENDIX A

Survey of Computer Programs for Heat Transfer (A.K. Noor)

The following tables and discussions were extracted from A.K. Noor's paper "Survey of Computer Programs for Heat Transfer Analysis."

These tables illustrate the solution capabilities, the heat transfer elements supported, the pre and post-processing options, the type of boundary conditions supported and any interfacing options for six of the heat transfer codes considered.

Table 1

Part I - Analytical Capabilities of the Program

		ANSYS	HEATING	MARC	NAS- TRAN	SINDA	THTD	
1.	Goal of Program System							
	General Purpose	•		•	•	•	•	
	Commercial	•		•	•	•		
	Research		•	•	•	•		
	Educational	•		•	•	•		
	Others (see program abstracts)							
2.	Method of Analysis							
	Finite Elements	•		•	•			
	Finite Differences		•			•	•	
	Boundary Integral Method				•			
	Perturbation Technique							
	Hybrid Analytical - Numerical Technique (see program abstracts)							
	Others (see program abstracts)					•		
3.	Space Dimensionality							
	Three-Dimensional	•	•	•	•	•	•	
	Two-Dimensional	•	•	•	•	•	•	
	One-Dimensional	•	•	•	•	•	•	

*Extracted from "Survey of Computer Programs for Heat Analysis" By A.K. Noor.

		ANSYS	HEATING	MARC	NAS- TRAN	SINDA	THID	
	Solids of Revolution	•		•	•			
	Boundary Elements				•			
	Scalar Elements			•	•			
	Point Contact Elements	•		•	•			
	Others (see program abstracts)							
4.	Range of Applications and Phenomena							
	Linear Steady State	•	•	•	•	•		
	Nonlinear Steady State	•	•	•	•	•	•	
	Thermal Frequencies and Mode Shapes							
	Linear Transient Response	•	•	•	•	•		
	Nonlinear Transient Response	•	•	•	•	•	•	
	Others (see program abstracts)							
5.	Formulation							
	a) Fundamental Unknowns							
	Temperatures	•	•			•	•	
	Temperatures and Flux							
	Enthalpy							
	Others (see program abstracts)					•		
	b) Elemental Matrices							
	Conduction	•	•			•	•	

		ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
6.	Capacitance	•		•	•		•	
	a) Consistent							
	b) Lumped	•	•	•	•	•		
	Convection	•		•	•	•	•	
	a) free							
	b) forced	•		•	•	•	•	
	Radiation	•		•	•	•	•	
	Interelement Convection and Radiation	•			•	•	•	
	User Supplied Elements (see program abstracts)				•			
	Others (see program abstracts)	•						
	Material Properties and Material Models							
	Isotropic	•	•	•	•	•	•	
	Anisotropic	•	•	•	•	•		
	Multilayered		•					
	Temperature-Dependent Properties	•	•	•	•	•	•	
		•	•	•		•	•	
		•	•		•	•		
		•	•		•	•		
		•	•		•	•	•	
	Perfect Conductors (Via Multipoint Constraints)	•	•	•	•	•	•	
				•	•			
					•			
		•	•	•				

			ANSYS	HEATING	MARC	NAS- TRAN	SINDA	THID	
	Latent Heat and Phase Change Effects		•		•		•	•	
	Material Added or Removed During Analysis				•		•		
	User Supplied (see program abstracts)				•		•		
	Others (see program abstracts)				•				
7.	Initial Conditions								
	Homogeneous.		•	•	•	•	•		
	Varying Throughout the Region			•	•	•	•	•	
	Initial Enthalpy (for Phase Change)					•			
	User-Supplied			•	•	•	•		
	Others (see program abstracts)								
8.	Boundary Conditions and Thermal Loads								
	Prescribed Temperatures	a) Steady State	•	•	•	•	•		
		b) Time Dependent	•	•	•	•	•	•	
	Thermal Flux Input	a) Steady	•	•	•	•	•		
		b) Temperature Dependent		•	•	•	•		
		c) Time Varying	•	•	•	•	•	•	
	Convection from a Surface to Its Surroundings	a) Steady State	•	•	•	•	•		
		b) Time Dependent	•	•	•	•	•	•	

			ANSYS	HEATING	MARC	NAS-TRAN	SINDA	THTD	
	Convection from Surroundings to a Surface	a) Steady State	•	•	•	•	•		
		b) Time Dependent	•	•	•	•	•	•	
Forced Convection			•		•	•	•		
	Radiation from a Surface to its Surroundings	a) Steady State	•	•	•	•	•		
		b) Time Dependent	•	•	•	•	•	•	
	Radiation from Surroundings to a Surface	a) Steady State	•	•	•	•	•		
		b) Time Dependent	•	•	•	•	•	•	
Radiation Between Narrow Gaps			•	•	•	•	•		
	Radiation Between n Surfaces with	a) User-Supplied View Factors	•	•	•	•	•		
		b) Internally Calculated View Factors				•		•	
Prescribed Fluid Flow			•			•	•		
Boundary Layer Convection					•	•			
	Volumetric Heat Generation	a) On Element Level	•		•	•		•	
		b) On Node Level		•	•	•	•		
Gap (Thermal Resistance)			•		•	•	•	•	
Boundary Conditions/Loads Added or Removed During Analysis			•		•	•	•		
Others (see program abstracts)									

				ANSYS	HEATING	MARC	NAS- TRAN	SINDA	THTD			
9.	Solution Techniques											
	Linear Steady State	a) Direct		•		•	•					
		b) Iterative			•			•				
		c) Others (see abstracts)					•					
	Nonlinear Steady State	a) Incremental				•						
		b) Iterative		•	•	•	•	•	•			
		c) Others (see abstracts)										
	Transient	a) Thermal Mode Superposition						•				
		b) Direct Integration	i) Explicit			•			•			
			ii) Implicit	User Specified Time Step	•	•	•	•	•	•		
				Automatic Time Step Selection				•	•			
			iii) Combined Explicit/Implicit									
		c) Finite Elements in the Time Domain										
		d) Moving - Deforming Grids										
		e) Others (see program abstracts)										
10.	Other Capabilities											
	Thermal Stress Analysis Capability	a) Uncoupled										
		b) Coupled										

		ANSYS	HEATING	MARC	NAS- TRAN	SINDA	THTD	
	Temperature Field Data Transmitted Directly from Heat Transfer Modules to Thermal Stress Modules	•		•	•		•	
	Enclosure Radiation with View Factor Calculation				•			
	Heat Input/Output at Constrained Boundaries	•		•	•		•	
	Cyclic Symmetry				•			
	Substructuring	•			•			
					•			
		•						
		•	•	•	•	•	•	
	Restart Capability				•			
	Others (see program abstracts)							
II.	Program Operational On	•	•	•	•	•	•	
	CDC	•	•	•	•	•		
	IBM	•		•	•	•	•	
	UNIVAC	•						
	Honeywell							
	Telefunken	•		•	•			
	AMDAIL							
	SIEMENS							
	ICL							

Part II - User Interface and Modeling Capabilities

		ANSYS	HEATING	NARC	NAS- TRAN	SINDA	THTD	
1.	Input Form and Sequence							
	a) Input Form							
	Fixed Format		•	•	•			
	Free Form-List Directed Format	•		•	•		•	
	Problem Oriented Language				•	•		
	Others (see program abstracts)							
	b) Input Sequence							
	User Directed	•		•			•	
	System Directed		•		•	•		
	User Supplied Subroutines (see program abstracts)			•	•	•		
2.	Model Generation and Checking							
	a) Automatic or Semi-Automatic Generator for:							
	Modal Point Coordinates	•	•	•	•		•	
	Element Connectivities	•	•	•	•		•	
	Constraints, Symmetry and Boundary Conditions	•	•	•	•		•	
	Substructure Connectivity				•			
	Repetition of Identical Segments	•		•	•		•	
	Others (see program abstracts)							

	ANSYS	HEAT- INGS	MARC	NAS- TRAN	SINDA	THTD	
b) Automatic or Semi-Automatic Generator for:	•		•	•			
One-Dimensional Elements	•		•	•			
Triangular Elements	•		•	•			
Quadrilateral Elements	•		•				
Body or Shell of Revolution Elements	•		•	•		•	
Three-Dimensional Solid Elements	•			•			
Two-Dimensional Shell Elements							
Curvilinear Finite Difference Grids							
Others (see program abstracts)							
c) Data Checking Facilities	•	•	•	•	•	•	
Line Printer	•	•	•	•	•	•	
Plotter							
Interactive Graphics							
Others (see program abstracts)							
d) Plots and Graphics Display of Model		•					
Complete Analysis Region	•	•	•	•		•	
Part of Analysis Region	•	•	•	•		•	
"Blow-Up" Option	•	•	•	•		•	

		ANSYS	HEAT- INGS	MARC	NAS- TRAN	SINDA	THID	
	Hidden Lines or Surfaces	•		•	•			
	Orthographic Views	•			•			
	Perspective and Isometric Views	•		•	•		•	
	Section View on Arbitrary Plane	•		•	•			
	Others (see program abstracts)							
	e) Other Facilities							
	Digitizer Input	•					•	
	Automatic Renumbering of Nodes, Elements or Equations	•		•	•		•	
	Table lookup of Data	•					•	
	Others (see program abstracts)							
3.	Results Output Form							
	a) Tabular Output							
	Fixed Set	•		•		•		
	User Defined Set and Sequences	•	•	•	•	•	•	
	Maximum and Minimum Quantities	•	•		•	•		
	Average and Maxima for Blocks of Nodes							
	Temperature or Flux Exceedances	•			•			
	Others (see program abstracts)							

		ANSYS	HEATING	MARC	NAS- TRAN	SINDA	THTD	
	b) File Output for User Post-Processing and Plotting	•	•	•	•	•		
	c) Plots							
	Isotherm Plots (Contours) of Temperatures/Flux	•	•	•	•		•	
	Surface Functions				•		•	
	Selective Output (e.g., by Elements or Regions)	•	•	•	•			
	Histories (e.g., Time History)	•	•	•	•	•	•	
	Others (see program abstracts)				•			
4.	Interactive Input and Control							
	Parameter Specification (e.g., Flux or Time Steps)	•					•	
	Singularity Check			•				
	Error Correction/Recovery	•						
	User Control of Matrix Decomposition							
	Others (see program abstracts)							

APPENDIX B

Survey of Structural Analysis Codes

















The following tables were extracted from a survey performed by R. Zirin of General Electric Gas Turbine Division.

These tables illustrate the solution capabilities, the element libraries, the types of loadings and the output features of the following structural codes: EASE2, STARDYNE, NASTRAN, ANSYS, MARC, SUPERB.

TYPES OF ANALYSIS

		PROGRAM					
ANALYTICAL CAPABILITY		EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
LINEAR STATICS	MECHANICAL LOADS	•	•	•	•	•	•
	TEMPERATURE LOADS	•	•	•	•	•	•
	EULER BUCKLING			•		•	
	INERTIA RELIEF			•			
DYNAMICS	MODE/FREQUENCY	•	•	•	•	•	✓
	FREQUENCY RESPONSE		•	•	•		
	TRANSIENT RESPONSE	•	•	•	•	•	
	SHOCK SPECTRA	•	•		•		✓
	RANDOM RESPONSE		•	•			
	NONLINEAR TRANSIENT			•	•	•	
NONLINEAR STATICS	NONLINEAR BUCKLING					•	
	LARGE DISPLACEMENT				•	•	
	PLASTICITY			•	•	•	
	CREEP				•	•	
	VISCOELASTICITY			•		•	
	LARGE STRAINS					•	
HEAT TRANSFER	STEADY STATE			•	•	•	•
	TRANSIENT			•	•	•	
SUBSTRUCTURES (SUPER-ELEMENTS)	STATIC		•	•	•		
	DYNAMIC		•	•	•		
	CYCLIC SYMMETRY			•			
MISCELLANEOUS	FRACTURE MECHANICS				•	•	
	FLUIDS			•	•	•	
	ELECTRIC CIRCUITS				•		
	OPTIMIZATION			•			
	ACOUSTIC CAVITIES			•			
	FATIGUE DAMAGE				•		















STRUCTURAL ANALYSIS ELEMENT/MATRIX LIBRARY

			PROGRAM					
			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPEHB
ELEMENT								
LINE ELEMENTS	ROD				•	•	•	
	BEAM		•	•	•	•	•	•
	TAPERED BEAM					•	•	
	OFFSET BEAM			•	•	•		•
	PINNED END BEAM		•	•	•			•
	CURVED BEAM						•	
FLAT MEMBRANES AND PLATES	3 NODE TRIANGLE		•	•	•	•	M	
	6 NODE TRIANGLE				M	•	M	C
	SHEAR PANEL				•			
	4 NODE QUAD		•	•	•	•	M	•
	8 NODE QUAD					S	M	S, C
CURVED SHELLS	3 NODE TRIANGLE						•	
	6 NODE TRIANGLE							C
	4 NODE QUAD						•	•
	8 NODE QUAD						•	S, C
	REDUCED THICK SHELL						•	

NOTES:

- M Membrane and/or plane strain only (no plate bending)
- S Includes sub-parametric forms with fewer nodes
- C Also includes cubic isoparametric element with two midside nodes

STRUCTURAL ANALYSIS ELEMENT/MATRIX LIBRARY (continued)

STRUCTURAL ANALYSIS ELEMENT/MATRIX LIBRARY (continued)				PROGRAM					
				EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
ELEMENT									
AXI-SYMMETRIC ELEMENTS	SHELLS	CONICAL				•	•	D	
		CURVED				•		•	
	TRIANGULAR RINGS	3 NODE				•	•	•	D
		6 NODE				•	•	D	C
	QUAD RINGS	4 NODE				•	•	•	•
		8 NODE					S	•	S, C
SOLID ELEMENTS	TETRA- HEDRON	4 NODE			•	•	•	D	
	WEDGES	6 NODE		•	•	•	•	D	D
		15 NODE						•	C
	HEXA- HEDRONS	8 NODE		•	•	•	•	•	•
		20 NODE				S		•	S, C
PIPE ELEMENTS		STRAIGHT		•	•	•	•	•	
		ELBOW		•	•		•	•	
		TEE			•				

NOTES:

- S Includes subparametric forms with fewer nodes
- C Also includes cubic isoparametric element with two midside nodes
- D Degenerate case














STRUCTURAL ANALYSIS ELEMENT/MATRIX LIBRARY (continued)

				PROGRAM					
ELEMENT				EASE2	STARDYNE	NASTRAN	ANSYS	MAHC	SUPERB
GENERAL STIFFNESS ELEMENTS	SPRING			1	•		•	•	•
	SCALAR SPRING					•			
	6 x 6 or 12 x 12 MATRIX				•	•	•		•
	GENERAL MATRIX					•			
MASSES	ELEMENT	LUMPED (DIAGONAL)		2	2	2	2		
		CONSISTENT				2	2	2	
	NON-STRUCTURAL	SCALAR (DOF)				•		•	
		NODAL		•	•	•	•		
		DISTRIBUTED				•			
		GUYAN REDUCTION		•	•	•	•		
		GENERAL MATRIX				•	•		
DAMPING	SCALAR					•			
	DASHPOT					•	•		
	DISCRETE VISCOUS $[C] = -[K] + \beta[M]$			•		•	•	•	
	STRUCTURAL $(1 + i\eta)[K]$					•			
	MODAL VISCOUS			•	•	•		•	
	GENERAL MATRIX					•	•		
OTHER ELEMENTS	GAP						•	•	
	FRICTION						•	•	
	RIGID			3	•	•			
	REBAR SOLID							•	
	ELASTIC FOUNDATION						•	•	
	CRACK TIP						•	•	•
	LAMINATED SHELL						•	•	•
	PLOT ONLY					•			•

NOTES:

- 1 See restraints
- 2 Generated from density
- 3 See constraints

HEAT TRANSFER-CONDUCTING ELEMENTS

ELEMENT			PROGRAM					
			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
LINEAR					•	•	•	•
PLANAR	3 NODE TRIANGLE				•	•	•	P,C
	4 NODE QUAD				•	•	•	•
	8 NODE QUAD					S	•	S,C
	TRANSVERSE CONDUCTING SHELL					•		
AXISYMMETRIC	TRIANGULAR RING				•	•	•	P,C
	4 NODE QUAD RING				•	•	•	•
	8 NODE QUAD RING					S	•	S,C
SOLID	TETRAHEDRON				•	•		
	WEDGE				•	•	D	
	8 NODE BRICK				•	•	•	•
	15 NODE WEDGE						D	C
	20 NODE BRICK						•	S,C
GENERAL MATRIX INPUT					•			

NOTES:

- S Contains subparametric forms with fewer number of nodes
- P Also contains parabolic isoparametric element with one midside node
- C Also contains cubic isoparametric element with two midside nodes
- D Degenerate case

COORDINATE SYSTEMS AND MATERIAL PROPERTIES

			PROGRAM					
FEATURE			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
COORDINATE SYSTEMS	BASIC (GLOBAL)	CARTESIAN	•	•	•	•	•	•
		CYLINDRICAL	•	•	•	•		•
		SPHERICAL			•	•		•
		GENERAL					1	
	SKEWED (LOCAL)	CARTESIAN	•	•	•	•	•	•
		CYLINDRICAL	•	•	•	•		•
		SPHERICAL			•	•		•
		GENERAL					1	
		MIXED	•	•	•	•	•	•
MATERIAL PROPERTIES		ISOTROPIC	•	•	•	•	•	•
		2-D ORTHOTROPIC		•	•	•	1	•
		3-D ORTHOTROPIC				•	1	•
		TEMPERATURE DEPENDENT	•		•	•	•	•
		STRESS DEPENDENT			•	•	•	
		TIME DEPENDENT				•	•	
		NONLINEAR ELASTIC					•	
	WORK HARDENING	ISOTROPIC				•	•	
		KINEMATIC				•	•	
		COMBINED				•	•	
		ORNL 10 CYCLE				•	•	
		GENERAL					1	

NOTES:

1 Performed by user subroutine

BOUNDARY CONDITIONS

BOUNDARY CONDITIONS			PROGRAM						
			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB	
FEATURE									
LOADING	STATIC	CONCENTRATED		•	•	•	•	•	•
		DISTRIBUTED (BEAM)		•	•	•	•	•	•
		PRESSURE	PLATES/SHELLS	•	•	•	•	•	•
			AXISYMMETRIC ELEMENTS			•	•	•	•
			SOLIDS	•	•	•	•	•	•
		TEMPERATURE		•	•	•	•	•	•
		ACCELERATION		•	•	•	•	•	•
		ROTATIONAL VELOCITY		•	•	•	•	1	•
		COMBINATION		•	•	•	•		+
		AXI-SYMMETRIC	AXISYMMETRIC SHELLS			•	•		
	AXISYMMETRIC RINGS					•			
	DYNAMIC	TIME DEPENDENT		•	•	•	•	•	
		FREQUENCY DEPENDENT			•	•	•		
		PSD RANDOM			•	•			
		SHOCK SPECTRUM		•	•		•		
DISPLACEMENT CONSTRAINTS		SINGLE POINT*		•	•	•	•	•	•
		MULTI POINT*		2		•	•	•	3
		SPECIFIED NONZERO DISPLACEMENT		•	•	•	•	•	•
HEAT TRANSFER		HEAT SOURCE/SINK				•	•	•	•
		CONVECTION				•	•	•	•
		RADIATION				•	•	•	
		SPECIFIED TEMPERATURE				•	•	•	•

NOTES: *Single point constraint is enforced zero translation(s) and/or rotation(s) in coordinate(s) associated with a node point
Multi-point constraint is enforced linear constraint relationships between translation(s) and/or rotation(s) which may be associated with different node points

- | | |
|---|--|
| 1 Applies to some elements | 3 Displacement components set equal on different nodes |
| 2 Specialized forms of rigid and interface coupling | + Stand alone program |

PRE- AND POST-PROCESSING

PRE- AND POST-PROCESSING			PROGRAM					
			EASE2	STARDYNE	NASTRAN	ANSYS	MARC	SUPERB
FEATURE								
PLOTTING	INPUT	UNDEFORMED GEOMETRY	+	•	•	•	•	•
		NODE LABELS	+	+	•	•	•	•
		ELEMENT LABELS	+		•	•	•	•
		PROPERTY LABELS			•	•		•
		2-D SECTIONS				•	•	
		BOUNDARY CONDITION LABELS	+		•			
		HIDDEN LINES REMOVED				✓	+	
	OUTPUT	DEFORMED GEOMETRY	+	•	•	•	•	•
		CONTOURS 2D STRUCTURE		+	•	•	•	•
		CONTOURS SOLID STRUCTURE				•	•	•
		TIME HISTORY	4	•	•	•	•	
		FREQUENCY RESPONSE		•	•,4	•		
		POWER SPECTRAL DENSITY		•	•,4			
ARBITRARY X VS. Y					•	+		
DATA GENERATION		NODES	1	1,2	1,2,3	1,2	2,3	1,2
		ELEMENTS	1	1	1,2,3	1,2	2,3	1,2
		RESTRAINTS	1	1	1,2	1	2,3	1
		LOADS	1	1	2	1	2,3	1
OUTPUT SORTING		BY LOAD CASES		•	•	•	•	•
		BY ELEMENT	•		•			
		MAX/MIN SUMMARY	•	•	•			•
		SELECTED NODES AND/OR ELEMENTS		•	•	•	•	
BANDWIDTH MINIMIZATION			•	•	•	W	•,W	•,W

NOTES:

- 1 Generates data in 1 "dimension"
- 2 Generates data in 2 "dimensions"
- 3 Generates data in 3 "dimensions"
- 4 Printer plots
- Stand alone program
- W Wavefront solution

APPENDIX C

Subroutine Prolog

This appendix lists the prologs for all of the subroutines used in the transfer module. The prologs represent a standard description of the purpose of the routine, the calling arguments and their attributes, the routines that are called, any commons that are used and special comments about the subroutine.

1. The first part of the document is a title page. It contains the title of the document, the author's name, and the date of the document.

FILE NAME	SUBROUTINE/ FUNCTION	PAGE NO
GE/TRANCITS	AREAD	17
GE/TRANCITS	CALCRD	12
GE/TRANCITS	CCNVEC	22
GE/TRANCITS	CHKPT	92
GE/TRANCITS	CHNODT	7
GE/TRANCITS	CINV	37
GE/TRANCITS	CKWIND	53
GE/TRANCITS	CORNCC	21
GE/TRANCITS	CTRAN	9
GE/TRANCITS	CWCRNT	31
GE/TRANCITS	DATINT	75
GE/TRANCITS	EINCOR	49
GE/TRANCITS	ELTRAN	98
GE/TRANCITS	EMINMX	40
GE/TRANCITS	ERRPRT	84
GE/TRANCITS	ETCORE	64
GE/TRANCITS	FCRNTP	34
GE/TRANCITS	FILREW	48
GE/TRANCITS	FNDELM	50
GE/TRANCITS	FNDJAC	94
GE/TRANCITS	FTIME	41
GE/TRANCITS	GENTMP	61
GE/TRANCITS	GETCRD	23
GE/TRANCITS	GETEWT	77
GE/TRANCITS	GETMPE	26
GE/TRANCITS	GETPD	13
GE/TRANCITS	GETWNO	52
GE/TRANCITS	GNLSUR	46
GE/TRANCITS	GPCOEF	101
GE/TRANCITS	GPTEMP	102
GE/TRANCITS	GTGAUS	19
GE/TRANCITS	GTNUMT	78
GE/TRANCITS	HTFIO	58
GE/TRANCITS	HTFPAS	67
GE/TRANCITS	HTICON	69
GE/TRANCITS	I3DSF	87
GE/TRANCITS	INITCM	99
GE/TRANCITS	JACBCK	96
GE/TRANCITS	KNTSPF	43
GE/TRANCITS	MARCRD	70
GE/TRANCITS	MARCTO	100
GE/TRANCITS	MAXTDF	35
GE/TRANCITS	MKRGF	54
GE/TRANCITS	MPERW	27
GE/TRANCITS	NTCORE	65
GE/TRANCITS	PARD	16
GE/TRANCITS	PDOFSF	93
GE/TRANCITS	PRCGEO	79
GE/TRANCITS	PRCSQF	76
GE/TRANCITS	PREPGM	8
GE/TRANCITS	PRNTIM	39

GE/TRANCITS		PROFAC	56
GE/TRANCITS		PUTWFI	24
GE/TRANCITS		RDCNTL	59
GE/TRANCITS		RDSPF	47
GE/TRANCITS		RDTEMP	30
GE/TRANCITS		RDTIME	4
GE/TRANCITS		REAARY	89
GE/TRANCITS		REACRD	20
GE/TRANCITS		READR	71
GE/TRANCITS		REMELM	11
GE/TRANCITS		RFILCR	44
GE/TRANCITS		RWCCF	25
GE/TRANCITS		RWELMS	2
GE/TRANCITS		RWNODS	1
GE/TRANCITS		RWSPCF	97
GE/TRANCITS		SECTION	10
GE/TRANCITS		SETFLP	29
GE/TRANCITS		SFDIST	91
GE/TRANCITS		SHAPEF	15
GE/TRANCITS		SHAPFV	95
GE/TRANCITS		SINTIT	73
GE/TRANCITS		SKIP	57
GE/TRANCITS		SKPREC	72
GE/TRANCITS		SORT	68
GE/TRANCITS		SORT2	90
GE/TRANCITS		SSURCH	42
GE/TRANCITS		STSTMP	62
GE/TRANCITS		TEMFOR	85
GE/TRANCITS		TFLSET	60
GE/TRANCITS		TIMCK	6
GE/TRANCITS		TPNAST	86
GE/TRANCITS		UNPAKF	33
GE/TRANCITS		VOLM	14
GE/TRANCITS		WRCRNT	66
GE/TRANCITS		WRITMP	36
GE/TRANCITS		WRSTMP	63
GE/TRANCITS		WRTDIR	81
GE/TRANCITS		WRTEMP	83
GE/TRANCITS		WRTGEO	82
GE/TRANCITS		WRTNTF	80
GE/TRANCITS		ZERDRF	45

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE= 1

```
2620C=====
2640      SUBROUTINE RWNODS(IREC,NN,XYZ,IRW)
2660C=====
2670CPR
2680CPR
2690CPR      ROUTINE--      RWNODS
2700CPR
2710CPR      DATE/WRITTEN BY--  8/04/82      RJ MAFFEO
2720CPR      DATE/REVISED BY--
2730CPR
2740CPR      FUNCTION/PURPOSE--
2750CPR      THIS I/O ROUTINE READS AND WRITES THE NODAL INFO
2760CPR      TO THE INTERNAL DATA BASE FILES
2770CPR      THIS FILE IS NOW IFL3
2780CPR      IN THE WRITE MODE IT AUTOMATICALLY ZEROS THIS FILE
2790CPR      FOR NODES NAMES THAT ARE NOT USED
2800CPR      THIS RANDOM FILE(IFL3) COULD EASILY BE REPLACED BY
2810CPR      AN ARRAY IF ENOUGH CORE WAS AVAILABEL
2820CPR
2830CPR      CALLING SEQUENCE-- CALL RWNODS(IREC,NN,XYZ,IRW)
2840CPR
2850CPR      CALLING ARGUMENTS--
2860CPR      NAME      ATTRIBUTES      DEFINITION
2870CPR      IREC      (I)      RECORD TO BE PROCESSED
2880CPR      NN      (I/O)      NODE NUMBER(NAME)
2890CPR      XYZ      (I/O) (3X1)  ARRAY OF COORDINATES
2900CPR      IRW      (I)      READ/WRITE SWITCH
2910CPR                      IRW=0---- READ
2920CPR                      IRW=1---- WRITE
2930CPR
2940CPR
2950CPR
2960CPR      FILES USED--
2970CPR          IFL3 --- RANDOM FILE TO STORE COORDINATES
2980CPR
2990CPR
3000CPR      COMMONS USED--
3010CPR          AFIL --- FILE CODE COMMON
3020CPR
3030CPR
3040CPR      FUNCTIONS/ROUTINES CALLED--
3050CPR          NONE
3060CPR
3070CPR      LIBRARIES ACCESSED--
3080CPR          NONE
3090CPR
3100CPR      LOCAL VARIABLES--
3110CPR          N/A
3120CPR
3130CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
3140CPR          NONE
3150CPR
3160CPR
3170CPR      SPECIAL REMARKS/INSTRUCTIONS--
3180CPR          HOST = H6000
3190CPR
3200CPR          *****
3630      END
```

```

3650C=====
3670      SUBROUTINE RWELMS(IREC,A,IRW,ICODE)
3690C=====
3700CPR
3710CPR
3720CPR      ROUTINE--    RWELMS
3730CPR
3740CPR      DATE/Written BY--  8/04/82    RJ MAFFEO
3750CPR      DATE/REVISED BY--
3760CPR
3770CPR      FUNCTION/PURPOSE--
3780CPR      THIS ROUTINE IS AN I/O ROUTINE
3790CPR      ITS PURPOSE IS TO READ/WRITE THE ELEMENT ATTRIBUTES TO THE
3800CPR      INTERNAL DATA BASE FILE
3810CPR      THIS DATA BASE FILE IS RANDOM FILE IFL4
3820CPR      THE ATTRIBUTES ARE
3830CPR              -- ELEMENT NAME
3840CPR              -- COORDINATES OF ALL 15 POINTS IN THE ELM
3850CPR              -- ELEMENT CONNECTIVITY OF THE 8 CORNERS
3860CPR              -- THE PACKED THTD FACE NUMBER
3870CPR              -- VOLUME OF ELEMENT
3880CPR
3890CPR      THE ROUTINE CAN PROCESS DIFFERENT AMOUNTS OF THESE
3900CPR      ATTRIBUTES BASED ON THE VALUE OF ICODE
3910CPR      THE RANDOM FILE IS NOT INITIALIZED PRESENTLY WITH THIS
3920CPR      ROUTINE
3930CPR
3940CPR
3950CPR      CALLING SEQUENCE--  CALL RWELMS(IREC,A,IRW,ICODE)
3960CPR
3970CPR      CALLING ARGUMENTS--
3980CPR          NAME      ATTRIBUTES      DEFINITION
3990CPR          IREC      (I)              RECORD NUMBER TO BE PROCESSED
4000CPR          A        (I/O) (56X1)    ARRAY OF ATTRIBUTES
4010CPR          IRW       (I)              READ/WRITE SWITCH
4020CPR                                      IRW=0---- READ
4030CPR                                      IRW=1---- WRITE
4040CPR          ICODE    (I)              SWITCH TO SET AMOUNT OF DATA TO PROCESS
4050CPR              ICODE=0 -- PROCESS ALL DATA
4060CPR              ICODE=1 -- PROCESS ELM NAME ONLY
4070CPR              ICODE=2 -- PROCESS ELM NAME AND CORNER COORDINATES
4090CPR              ICODE=4 -- PROCESS EVERYTHING EXCEPT VOLUME
4100CPR              ICODE=5 -- PROCESS ALL DATA
4110CPR
4120CPR
4130CPR
4140CPR      FILES USED--
4150CPR          IFL4 -- RANDOM FILE TO STORE ELM ATTRIBUTES
4160CPR
4170CPR
4180CPR      COMMONS USED--
4190CPR          AFIL -- COMMON TO DEFINE FILECODES
4200CPR
4210CPR
4220CPR
4230CPR      FUNCTIONS/ROUTINES CALLED--
4240CPR          NONE
4250CPR
4260CPR
4270CPR      LIBRARIES ACCESSED--
4280CPR          NONE
4290CPR
4300CPR      LOCAL VARIABLES--
4310CPR          N/A
4320CPR
4330CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
4340CPR          NONE

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE= 3

4350CPR

4360CPR

4370CPR SPECIAL REMARKS/INSTRUCTIONS--

4380CPR HOST = H6000

4390CPR

4400CPR *****

4850 END

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE= 4

```
4870C=====
4890      SUBROUTINE RDTIME
4920C=====
4930CPR
4940CPR
4950CPR      ROUTINE--      RDTIME
4960CPR
4970CPR      DATE/WRITTEN BY--  8/18/82      RJ MAFFEO
4980CPR      DATE/REVISED BY-- 9/13/83      SR MARTIN
4990CPR          REASON      --  CONVERT FROM F4BT FILES TO THE NEW
5000CPR                      HEAT XFER OUTPUT FILE FORM
5010CPR
5020CPR      FUNCTION/PURPOSE--
5030CPR
5040CPR      THIS ROUTINE READS THE TIME VALUES FROM THE HEAT XFER OUTPUT FILE
5050CPR      AND DETERMINES IF THEY ARE DESIRED FOR THIS RUN  IT
5060CPR      AUTOMATICALLY ELIMINATES THE NON-STP TIME STEPS  IT ALSO
5070CPR      CALLS THE REQUIRED ROUTINE TO PERFORM INITIAL CHECKS ON
5080CPR      TIMES AND ELM COUNTS
5090CPR      THE HEAT TRANSFR OUTPUT FILE IS REWOUND BEFORE RETURNING
5100CPR      FROM THIS ROUTINE
5110CPR
5120CPR
5130CPR      CALLING SEQUENCE-- CALL RDTIME
5140CPR
5150CPR      CALLING ARGUMENTS--
5160CPR          NAME      ATTRIBUTES      DEFINITION
5170CPR
5180CPR      FILES USED--
5190CPR
5200CPR          INTITS -- HEAT XFER OUTPUT FILE (TITAN INPUT)
5210CPR          IERF  -- ERROR REMARK FILE
5220CPR          INOT6 -- HARD COPY FILE
5230CPR
5240CPR
5250CPR      COMMONS USED--
5260CPR
5270CPR          AFIL      -- FILE CODES
5280CPR          SIZE      -- PROBLEM SIZE PARAMETERS
5290CPR          CNTLTM    -- TIME/TEMPERATURE CONTROL PARAMETERS
5300CPR
5310CPR
5320CPR
5330CPR      FUNCTIONS/ROUTINES CALLED--
5340CPR
5350CPR          TIMCK
5360CPR          CHNODT
5370CPR
5380CPR
5390CPR      LIBRARIES ACCESSED--
5400CPR          NONE
5410CPR
5420CPR
5430CPR      LOCAL VARIABLES--
5440CPR
5450CPR          NSTPTM    -- COUNT OF STP TIMES PROCESSED
5460CPR          NCNTT     -- COUNT OF TOTAL TIMES PROCESSED
5470CPR          NTMDO     -- USER SPECIFIED NUMBER OF STP TIMES TO DO
5480CPR          TIMT      -- ARRAY OF TIMES PROCESSED  THIS ARRAY WILL
5490CPR                      -- BE USED AS A DIRECTORY TO POINT TO THE
5500CPR                      -- LOCATION ON THE HEAT XFER OUTPUT FILE FOR EACH TIME
5510CPR
5520CPR
5530CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
5540CPR
5550CPR
5560CPR      SPECIAL REMARKS/INSTRUCTIONS--
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE= 5

557OCPR

HOST = H6000

558OCPR

559OCPR

6640

END

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE= 6

```
6660C=====
6680      SUBROUTINE TIMCK(TIME,ITIME)
6710C=====
6720CPR
6730CPR
6740CPR      ROUTINE--    TIMCK
6750CPR
6760CPR      DATE/WRITTEN BY--  8/18/82    RJ MAFFEO
6770CPR      DATE/REVISED BY--
6780CPR
6790CPR      FUNCTION/PURPOSE--
6800CPR
6810CPR      THIS ROUTINE CHECKS TO SEE IF THE PRESENT TIME VALUE
6820CPR      IS ONE OF THE TIMES DESIRED
6830CPR
6840CPR
6850CPR      CALLING SEQUENCE--    CALL TIMCK(TIME,ITIME)
6860CPR
6870CPR      CALLING ARGUMENTS--
6880CPR          NAME      ATTRIBUTES      DEFINITION
6890CPR
6900CPR          TIME      (1)      PRESENT TIME VALUE OF TIME STEP
6910CPR          ITIME      (0)      DESIRED SWITCH
6920CPR                                ITIME=0 >>> NOT DESIRED
6930CPR                                ITIME=1 >>> DESIRED
6940CPR
6950CPR
6960CPR
6970CPR      FILES USED--    NONE
6980CPR
6990CPR      COMMONS USED--
7000CPR
7010CPR      CNTLTM    --- TIME/TEMPERATURE CONTROL PARAMATERS
7020CPR
7030CPR
7040CPR
7050CPR      FUNCTIONS/ROUTINES CALLED--    NONE
7060CPR
7070CPR      LIBRARIES ACCESSED--
7080CPR      NONE
7090CPR
7100CPR
7110CPR      LOCAL VARIABLES--
7120CPR
7130CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
7140CPR
7150CPR
7160CPR      SPECIAL REMARKS/INSTRUCTIONS--
7170CPR          HOST = H6000
7180CPR
7190CPR      *****
7840      END
```

TITLE= DATE =07/15/85 TIME = 23 42 PAGE= 7

```
7860C=====
7880      SUBROUTINE CHNODT
7910C=====
7920CPR
7930CPR
7940CPR      ROUTINE--      CHNODT
7950CPR
7960CPR      DATE/WRITTEN BY--  8/18/82    RJ MAFFEO
7970CPR      DATE/REVISED BY--
7980CPR
7990CPR      FUNCTION/PURPOSE--
8000CPR
8010CPR      THIS ROUTINE MAKES TWO BASIC CHECKS ON THE INPUT
8020CPR      IT CHECKS TO SEE THAT THE NUMBER OF ELEMENTS IN EACH TIMESTEP
8030CPR      IS GREATER THAN OR EQUAL TO THE NUMBER OF GEOMETRY ELEMENTS
8040CPR      IT ALSO CHECKS TO INSURE THAT ALL ELMS ON THE
8050CPR      NEUTRAL ELM FILE HAVE TEMPERATURES ON THE THT
8060CPR      OUTPUT FILE
8070CPR
8080CPR
8090CPR      CALLING SEQUENCE--      CALL CHNODT
8100CPR
8110CPR      CALLING ARGUMENTS--
8120CPR          NAME      ATTRIBUTES      DEFINITION
8130CPR
8140CPR
8150CPR
8160CPR      FILES USED--
8170CPR
8180CPR      IFLO    -- THT OUTPUT FILE
8190CPR      IERF    -- ERROR REMARK FILE
8200CPR
8210CPR      COMMONS USED--
8220CPR
8230CPR          AFIL
8240CPR          WORK
8250CPR          SIZE
8260CPR          ELMDAT
8270CPR          CNTLFL
8280CPR
8290CPR
8300CPR
8310CPR      FUNCTIONS/ROUTINES CALLED--
8320CPR
8330CPR          RDMANY
8340CPR          RWELMS
8350CPR
8360CPR      LIBRARIES ACCESSED--
8370CPR          NONE
8380CPR
8390CPR
8400CPR      LOCAL VARIABLES--
8410CPR
8420CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
8430CPR
8440CPR
8450CPR      SPECIAL REMARKS/INSTRUCTIONS--
8460CPR          HOST = H6000
8470CPR
8480CPR      *****
9510      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE= 8

```
9530C=====
9550      SUBROUTINE PREPGM
9580C=====
9590CPR
9600CPR
9610CPR      ROUTINE--    PREPGM
9620CPR
9630CPR      DATE/WRITTEN BY--  8/18/82    RJ MAFFEO
9640CPR      DATE/REVISED BY--
9650CPR
9660CPR      FUNCTION/PURPOSE--
9670CPR
9680CPR      THIS ROUTINE CALLS ALL ROUTINES NEEDED TO
9690CPR      TRANSFORM,SECTION (WINDOW) AND/OR
9700CPR      ELIMINATE UNWANTED ELMS FROM THE PROBLEM
9710CPR
9720CPR
9730CPR      CALLING SEQUENCE--    CALL PREPGM
9740CPR
9750CPR      CALLING ARGUMENTS--    NONE
9760CPR          NAME      ATTRIBUTES      DEFINITION
9770CPR          NONE      NONE            NONE
9780CPR
9790CPR      FILES USED--    NONE
9800CPR
9810CPR      COMMONS USED--
9820CPR
9830CPR          SIZE
9840CPR          ELMDAT
9850CPR          CNTLGM
9860CPR
9870CPR
9880CPR
9890CPR      FUNCTIONS/ROUTINES CALLED--
9900CPR
9910CPR          RWELMS
9920CPR          CTRAN
9930CPR          SECTION
9940CPR          REMELM
9950CPR
9960CPR
9970CPR      LIBRARIES ACCESSED--
9980CPR          NONE
9990CPR
10000CPR
10010CPR      LOCAL VARIABLES--
10020CPR
10030CPR          JSLC  -- WINDOW SWITCH
10040CPR                  JSLC=1 >>> ELM IS IN WINDOW
10050CPR                  JSLC=-1 >>> ELM IS NOT IN WINDOW
10060CPR
10070CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
10080CPR
10090CPR
10100CPR      SPECIAL REMARKS/INSTRUCTIONS--
10110CPR          HOST = H6000
10120CPR
10130CPR          *****
10750      END
```


TITLE= DATE =07/15/85 TIME = 23 42 PAGE= 9

```
10770C=====
10790      SUBROUTINE CTRAN(XO,YO,ZO,C)
10820C=====
10830CPR
10840CPR
10850CPR      ROUTINE--    CTRAN
10860CPR
10870CPR      DATE/WRITTEN BY--  8/18/82    RJ MAFFED
10880CPR      DATE/REVISED BY--
10890CPR
10900CPR      FUNCTION/PURPOSE--
10910CPR
10920CPR      THIS ROUTINE TRANSFORMS THE COORDINATES OF THE EIGHT
10930CPR      CORNER POINTS OF EACH ELM
10940CPR      TRANSFORMS ARE LIMITED TO TRANSLATIONS
10950CPR
10960CPR
10970CPR      CALLING SEQUENCE--    CALL CTRAN(XO,YO,ZO,C)
10980CPR
10990CPR      CALLING ARGUMENTS--
11000CPR          NAME      ATTRIBUTES      DEFINITION
11010CPR          XO        (I)              X OFFSET
11020CPR          YO        (I)              Y OFFSET
11030CPR          ZO        (I)              Z OFFSET
11040CPR          C          (I/O) ARRAY      COORDINATE ARRAY
11050CPR
11060CPR
11070CPR
11080CPR      FILES USED--    NONE
11090CPR
11100CPR      COMMONS USED--    NONE
11110CPR
11120CPR
11130CPR      FUNCTIONS/ROUTINES CALLED--    NONE
11140CPR
11150CPR      LIBRARIES ACCESSED--
11160CPR          NONE
11170CPR
11180CPR
11190CPR      LOCAL VARIABLES--
11200CPR
11210CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
11220CPR
11230CPR
11240CPR      SPECIAL REMARKS/INSTRUCTIONS--
11250CPR          HOST = H6000
11260CPR
11270CPR      *****
11440      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=10

```

1146OC=====
1148O      SUBROUTINE SCTION(C,XMN,YMN,ZMN,XX,YY,ZZ,INOT)
1151OC=====
1152OCPR
1153OCPR
1154OCPR      ROUTINE--      SCTION
1155OCPR
1156OCPR      DATE/WRITTEN BY--  8/18/82    RJ MAFFEO
1157OCPR      DATE/REVISED BY--
1158OCPR
1159OCPR      FUNCTION/PURPOSE--
1160OCPR
1161OCPR      THIS ROUTINE CHECKS TO SEE IF ANY OF THE CORNER
1162OCPR      COORDINATES ARE OUTSIDE OF THE WINDOW SPECIFIED
1163OCPR      BY THE MIN AND MAX VALUES OF X,Y,Z
1164OCPR      NOTE IF ANY POINT OF AN ELM IS OUTSIDE THE WINDOW
1165OCPR      THE ELM IS CONSIDERED OUT OF THE WINDOW.
1166OCPR
1167OCPR
1168OCPR
1169OCPR      CALLING ARGUMENTS--
1170OCPR      NAME      ATTRIBUTES      DEFINITION
1171OCPR
1172OCPR      C          (I)          COORDINATE ARRAY
1173OCPR      XMN >
1174OCPR      YMN >>   (I)          MIN VALUE OF X,Y,Z FOR WINDOW
1175OCPR      ZMN >
1176OCPR      XX >
1177OCPR      YY >>   (I)          MAX VALUE OF X,Y,Z FOR WINDOW
1178OCPR      ZZ >
1179OCPR
1180OCPR      INOT      (O)          IN/OUT SWITCH
1181OCPR      INOT=1 >> ELM IS IN THE WINDOW
1182OCPR      INOT=-1 >> ELM IS OUT OF WINDOW
1183OCPR
1184OCPR
1185OCPR
1186OCPR      FILES USED--      NONE
1187OCPR
1188OCPR      COMMONS USED--     NONE
1189OCPR
1190OCPR
1191OCPR      FUNCTIONS/ROUTINES CALLED--  NONE
1192OCPR
1193OCPR      LIBRARIES ACCESSED--
1194OCPR      NONE
1195OCPR
1196OCPR
1197OCPR      LOCAL VARIABLES--
1198OCPR
1199OCPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
1200OCPR
1201OCPR
1202OCPR      SPECIAL REMARKS/INSTRUCTIONS--
1203OCPR      HOST = H6000
1204OCPR
1205OCPR      *****
1236O      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=11

```
12380C=====
12400      SUBROUTINE REMELM
12420C=====
12440CPR
12450CPR=====
12460CPR
12470CPR
12480CPR      ROUTINE--          REMELM
12490CPR
12500CPR      DATE/WRITTEN BY-- 09/13/82      RJ MAFFEO
12510CPR      DATE/REVISED BY--
12520CPR
12530CPR      FUNCTION/PURPOSE--
12540CPR
12550CPR      THIS SUB ELIMINATES UNDESIRE ELMS FROM THE RANDOM
12560CPR      ELM GEOM FILE
12570CPR
12580CPR
12590CPR
12600CPR      CALLING ARGUMENTS --
12610CPR          NAME          ATTRIBUTES      DEFINITION
12620CPR
12630CPR
12640CPR      COMMONS USED --
12650CPR
12660CPR          ELMDAT
12670CPR          CNTLGM
12680CPR
12690CPR      FUNCTIONS/ROUTINES CALLED --
12700CPR
12710CPR          RWELMS
12720CPR
12730CPR      FILES USED --
12740CPR
12750CPR
12760CPR      LIBRARIES ACCESSED --
12770CPR      NONE
12780CPR
12790CPR
12800CPR      LOCAL VARIABLES --
12810CPR
12820CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
12830CPR
12840CPR
12850CPR      SPECIAL REMARKS/INSTRUCTIONS --
12860CPR          HOST = H6000
12870CPR
12880CPR
12890CPR=====
13470      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=12

```

13490C=====
13510      SUBROUTINE CALCRD
13530C=====
13540CPR
13550CPR
13560CPR      ROUTINE--    CALCRD
13570CPR
13580CPR      DATE/WRITTEN BY--  8/08/82    RJ MAFFEO
13590CPR      DATE/REVISED BY--
13600CPR
13610CPR      FUNCTION/PURPOSE--
13620CPR
13630CPR      THE PURPOSE OF THIS ROUTINE IS TO CALL ALL
13640CPR      REQUIRED ROUTINES TO COMPUTE THE COORDINATES OF THE
13650CPR      CENTROID OF THE ELMS AND THE COORDINATES OF THE
13660CPR      FACE CENTERS IT THEN CALLS THE ELM I/O ROUTINE TO STORE
13670CPR      THIS INFO INTO THE INTERNAL DATA BASE
13680CPR
13690CPR
13700CPR      CALLING SEQUENCE--    CALL CALCRD
13710CPR
13720CPR      CALLING ARGUMENTS--      NONE
13730CPR
13740CPR          NAME      ATTRIBUTES      DEFINITION
13750CPR          NONE      NONE            NONE
13760CPR
13770CPR      FILES USED--
13780CPR
13790CPR          IERF      -- ERROR REMARK FILE
13800CPR
13810CPR      COMMONS USED--
13820CPR
13830CPR          AFIL      -- FILE CODE COMMON
13840CPR          ELMDAT   -- ELM ATTRIBUTE COMMON
13850CPR          SIZE      -- PROBLEM SIZE COMMON
13860CPR          WORK      -- SCRATCH WORK SPACE COMMON
13870CPR          CRNDEF   -- LOCAL COMMON TO COORD CALC  ROUTINES
13880CPR
13890CPR      FUNCTIONS/ROUTINES CALLED--
13900CPR
13910CPR          GETPD
13920CPR          RWELMS
13930CPR          VOLM
13940CPR          AREAD
13950CPR          REACRD
13960CPR
13970CPR      LIBRARIES ACCESSED--
13980CPR          NONE
13990CPR
14000CPR      LOCAL VARIABLES--
14010CPR
14020CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
14030CPR
14040CPR      MANY OF THE ROUTINES CALLED BY CALCRD ARE ALSO USED
14050CPR      IN SIGMA THTD DECK GENERATOR THEY CAN COMPUTE MUCH MORE THAN
14060CPR      JUST COORDINATES(SUCH AS VOLUMES AREAS AND DELTA X).
14070CPR
14080CPR
14090CPR
14100CPR      SPECIAL REMARKS/INSTRUCTIONS--
14110CPR          HOST = H6000
14120CPR
14130CPR          *****
15130      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=13

```

15150C=====
15170      SUBROUTINE GETPD(P3D,P2D)
15190C=====
15210CPR
15220CPR=====
15230CPR
15240CPR
15250CPR      ROUTINE--          GETPD
15260CPR
15270CPR      DATE/WRITTEN BY-- 09/13/82      RJ MAFFEO
15280CPR      DATE/REVISED BY--
15290CPR
15300CPR      FUNCTION/PURPOSE--
15310CPR
15320CPR      THIS SUB COMPUTES AND STORES ALL OF THE PARTIAL DERIVATIVES
15330CPR      OF THE SHAPE FUNCTIONS.BOTH 3D AND 2D
15340CPR
15350CPR
15360CPR      CALLING ARGUMENTS --
15370CPR          NAME          ATTRIBUTES      DEFINITION
15380CPR
15390CPR          P3D            (0)              3D PARTIAL DERIVATIVES
15400CPR          P2D            (0)              2D PARTIAL DERIVATIVES
15410CPR
15420CPR      COMMONS USED --
15430CPR
15440CPR          CRNDEF --- SEE CALCRD
15450CPR
15460CPR      FUNCTIONS/ROUTINES CALLED --
15470CPR
15480CPR          PARD
15490CPR          GTGAUS
15500CPR
15510CPR      FILES USED --
15520CPR
15530CPR
15540CPR      LIBRARIES ACCESSED --
15550CPR      NONE
15560CPR
15570CPR
15580CPR      LOCAL VARIABLES --
15590CPR
15600CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
15610CPR
15620CPR
15630CPR      SPECIAL REMARKS/INSTRUCTIONS --
15640CPR          HOST = H6000
15650CPR
15660CPR
15670CPR=====
16020      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=14

```

16040C=====
16060      SUBROUTINE VOLM(X,Y,Z,GP,P3D,VOL,XC,YC,ZC,Q,IVER)
16080C=====
16100CPR
16110CPR=====
16120CPR
16130CPR
16140CPR      ROUTINE--          VOLM
16150CPR
16160CPR      DATE/WRITTEN BY-- 09/13/82      RJ MAFFEO
16170CPR      DATE/REVISED BY--
16180CPR
16190CPR      FUNCTION/PURPOSE--
16200CPR
16210CPR      THIS SUB DOES NUMERICAL INTEGRATION FOR VOL,XC,YC,ZC
16220CPR      IT COMPUTES DETJ AND THE JACOBIAN
16230CPR
16240CPR
16250CPR
16260CPR      CALLING ARGUMENTS --
16270CPR          NAME          ATTRIBUTES      DEFINITION
16280CPR
16290CPR          X              (I)              ARRAY OF X COORDS
16300CPR          Y              (I)              ' ' Y '
16310CPR          Z              (I)              ' ' Z '
16320CPR          GP              (I)              ARRAY OF GAUSS POINTS
16330CPR          P3D              (I)              3D PARTIAL DERIVATIVES
16340CPR          VOL              (O)              VOLUME
16350CPR          XC              (O)              X COORD OF CENTROID
16360CPR          YC              (O)              Y '
16370CPR          ZC              (O)              Z '
16380CPR          Q              (O)              JACOBIAN
16390CPR          IVER              (O)              IVER=1 > ZERO VOLUME ELM
16400CPR
16410CPR      COMMONS USED --
16420CPR
16430CPR
16440CPR      FUNCTIONS/ROUTINES CALLED --
16450CPR
16460CPR          SHAPEF
16470CPR
16480CPR      FILES USED --
16490CPR
16500CPR
16510CPR      LIBRARIES ACCESSED --
16520CPR      NONE
16530CPR
16540CPR
16550CPR      LOCAL VARIABLES --
16560CPR
16570CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
16580CPR
16590CPR
16600CPR      SPECIAL REMARKS/INSTRUCTIONS --
16610CPR          HOST = H6000
16620CPR
16630CPR
16640CPR=====
17150      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=15

```

17170CPR=====
17190      SUBROUTINE SHAPEF(X,Y,Z,TETA,YNETA,ZETA,XP,YP,ZP)
17210CPR=====
17230CPR
17240CPR=====
17250CPR
17260CPR
17270CPR      ROUTINE--          SHAPEF
17280CPR
17290CPR      DATE/WRITTEN BY-- 09/14/82      RJ MAFFEO
17300CPR      DATE/REVISED BY--
17310CPR
17320CPR      FUNCTION/PURPOSE--
17330CPR
17340CPR      THIS SUB COMPUTES THE VALUES OF X,Y,Z AT THE GUASS POINTS
17350CPR      BASED ON LINEAR 3D SHAPE FUNCTIONS
17360CPR
17370CPR
17380CPR
17390CPR
17400CPR      CALLING ARGUMENTS --
17410CPR      NAME          ATTRIBUTES      DEFINITION
17420CPR
17430CPR      X              (I) >
17440CPR      Y              (I) >>      ARRAYS OF COORDINATES
17450CPR      Z              (I) >
17460CPR      TETA           (I) >
17470CPR      YNETA          (I) >>      LOCAL COORDINATES
17480CPR      ZETA           (I) >
17490CPR      XP             (O) >
17500CPR      YP             (O) >>      DESIRED GLOBAL COORDINATES
17510CPR      ZP             (O) >
17520CPR
17530CPR      COMMONS USED --
17540CPR
17550CPR      CRNDEF
17560CPR
17570CPR      FUNCTIONS/ROUTINES CALLED --
17580CPR
17590CPR
17600CPR      FILES USED --
17610CPR
17620CPR
17630CPR      LIBRARIES ACCESSED --
17640CPR      NONE
17650CPR
17660CPR
17670CPR      LOCAL VARIABLES --
17680CPR
17690CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
17700CPR
17710CPR
17720CPR      SPECIAL REMARKS/INSTRUCTIONS --
17730CPR      HOST = H6000
17740CPR
17750CPR
17760CPR=====
17950      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=16

```

17970C=====
17990      SUBROUTINE PARD(TETA,YNETA,ZETA,IGP,IND,PD)
18010C=====
18030CPR
18040CPR=====
18050CPR
18060CPR
18070CPR      ROUTINE--          PARD
18080CPR
18090CPR      DATE/WRITTEN BY-- 09/14/82      RJ MAFFEO
18100CPR      DATE/REVISED BY--
18110CPR
18120CPR      FUNCTION/PURPOSE--
18130CPR
18140CPR      THIS SUB COMPUTES THE DERIVATIVES OF THE
18150CPR      SHAPE FUNCTIONS WRT THE LOCAL SYSTEM
18160CPR      BOTH FOR VOLUMES(3D) AND AREAS(2D)
18170CPR
18180CPR
18190CPR
18200CPR      CALLING ARGUMENTS --
18210CPR          NAME          ATTRIBUTES      DEFINITION
18220CPR
18230CPR          TETA          (I) >
18240CPR          YNETA         (I) >>          LOCAL COORDINATES
18250CPR          ZETA          (I) >
18260CPR          IGP            (I)              GAUSS POINT COUNTER
18270CPR          IND            (I)              2D/3D INDICATOR
18280CPR                                          IND=0 >> 3D
18290CPR                                          IND=1 THRU 6 >> 2D IND=FACE NUMBER
18300CPR          PD            (O)              ARRAY OF DERIVATIVES
18310CPR
18320CPR      COMMONS USED --
18330CPR
18340CPR          CRNDEF
18350CPR
18360CPR      FUNCTIONS/ROUTINES CALLED --
18370CPR
18380CPR
18390CPR      FILES USED --
18400CPR
18410CPR
18420CPR      LIBRARIES ACCESSED --
18430CPR      NONE
18440CPR
18450CPR
18460CPR      LOCAL VARIABLES --
18470CPR
18480CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
18490CPR
18500CPR
18510CPR      SPECIAL REMARKS/INSTRUCTIONS --
18520CPR          HOST = H6000
18530CPR
18540CPR
18550CPR=====
18950      END

```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=17

```

18970C=====
18990      SUBROUTINE AREAD(X,Y,Z,P2D,Q,ARA,DX2,
19020C=====
19040CPR
19050CPR=====
19060CPR
19070CPR
19080CPR      ROUTINE--          AREAD
19090CPR
19100CPR      DATE/WRITTEN BY-- 09/14/82      RJ MAFFEO
19110CPR      DATE/REVISED BY--
19120CPR
19130CPR      FUNCTION/PURPOSE--
19140CPR
19150CPR      THIS SUB COMPUTES THE FACE AREAS AND DELTA X
19160CPR      FOR 3D THT ELMS
19170CPR      NOTE IT COMPUTES DELTA X TWO WAYS
19180CPR      IDX=1      DX=DIST FROM ELM CENTROID TO FACE CENTROID
19190CPR      IDX=2      DX=PERPENDICULAR DIST FROM ELM CENT TO FACE
19200CPR      IDX=3      SWITCH TO JUST CALC CENTROID OF FACE 5
19210CPR
19220CPR      IT IS BEING USED HERE TO COMPUTE THE COORDINATES
19230CPR      OF THE FACE CENTERS
19240CPR
19250CPR
19260CPR
19270CPR
19280CPR      CALLING ARGUMENTS --
19290CPR      NAME          ATTRIBUTES      DEFINITION
19300CPR
19310CPR      X              (I) >
19320CPR      Y              (I) >>      ARRAY OF CORNER COORDS
19330CPR      Z              (I) >
19340CPR      P2D           (I)
19350CPR      Q              (O)
19360CPR      ARA            (O)
19370CPR      DX2           (O)
19380CPR      IDX          (I)
19390CPR      XC            (I) >
19400CPR      YC            (I) >>      COORDS OF VOLUME CENTROID
19410CPR      ZC            (I) >
19420CPR      XA            (O) >
19430CPR      YA            (O) >>      ARRAYS OF COORDS FOR FACE CENTERS
19440CPR      ZA            (O) >
19450CPR
19460CPR      COMMONS USED --
19470CPR
19480CPR      CRNDEF
19490CPR
19500CPR      FUNCTIONS/ROUTINES CALLED --
19510CPR
19520CPR      GTGAUS
19530CPR      SHAPEF
19540CPR
19550CPR      FILES USED --
19560CPR
19570CPR
19580CPR      LIBRARIES ACCESSED --
19590CPR      NONE
19600CPR
19610CPR
19620CPR      LOCAL VARIABLES --
19630CPR
19640CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
19650CPR
19660CPR
19670CPR      SPECIAL REMARKS/INSTRUCTIONS --

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=18

19680CPR HOST = H6000

19690CPR

19700CPR

19710CPR=====

20780 END

TITLE= DATE =07/15/85 TIME = 23 42 PAGE=19

```

20800C=====
20820      SUBROUTINE GTGAUS(IND,I1,I2,I3,IP)
20840C=====
20860CPR
20870CPR=====
20880CPR
20890CPR
20900CPR      ROUTINE--          GTGAUS
20910CPR
20920CPR      DATE/WRITTEN BY-- 09/14/82      RJ MAFFED
20930CPR      DATE/REVISED BY--
20940CPR
20950CPR      FUNCTION/PURPOSE--
20960CPR
20970CPR      THIS SUB GETS APPROPRIATE GAUSS POINTS
20980CPR      FOR EACH FACE OF THE 3D BRICK
20990CPR
21000CPR
21010CPR
21020CPR
21030CPR      CALLING ARGUMENTS --
21040CPR          NAME          ATTRIBUTES          DEFINITION
21050CPR
21060CPR          IND          (I)          FACE INDICATOR
21070CPR          I1          (O) >
21080CPR          I2          (O) >
21090CPR          I3          (O) >>          GAUSS POINT COUNTERS
21100CPR          IP          (O) >
21110CPR
21120CPR      COMMONS USED --
21130CPR
21140CPR
21150CPR      FUNCTIONS/ROUTINES CALLED --
21160CPR
21170CPR
21180CPR      FILES USED --
21190CPR
21200CPR
21210CPR      LIBRARIES ACCESSED --
21220CPR      NONE
21230CPR
21240CPR
21250CPR      LOCAL VARIABLES --
21260CPR
21270CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
21280CPR
21290CPR
21300CPR      SPECIAL REMARKS/INSTRUCTIONS --
21310CPR          HOST = H6000
21320CPR
21330CPR
21340CPR=====
21450      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=20

```

21470C=====
21490      SUBROUTINE REACRD(CRD,XC,YC,ZC,XA,YA,ZA)
21510C=====
21530CPR
21540CPR=====
21550CPR
21560CPR
21570CPR      ROUTINE--          REACRD
21580CPR
21590CPR      DATE/WRITTEN BY-- 09/15/82      RJ MAFFEO
21600CPR      DATE/REVISED BY--
21610CPR
21620CPR      FUNCTION/PURPOSE--
21630CPR
21640CPR      THIS SUB REARRANGES THE FACE CENTER COORDINATE ARRAYS
21650CPR      AND THE VOLUME CENTROIDS INTO THE CRD ARRAY
21660CPR
21670CPR
21680CPR
21690CPR
21700CPR      CALLING ARGUMENTS --
21710CPR          NAME          ATTRIBUTES      DEFINITION
21720CPR
21730CPR          CRD            (O)              BRICK COORD ARRAY
21740CPR          XC             (I)  >
21750CPR          YC             (I)  >>          VOLUME CENTROIDS
21760CPR          ZC             (I)  >
21770CPR          XA             (I)  >
21780CPR          YA             (I)  >>          FACE CENTER ARRAYS
21790CPR          ZA             (I)  >
21800CPR
21810CPR      COMMONS USED --
21820CPR
21830CPR
21840CPR      FUNCTIONS/ROUTINES CALLED --
21850CPR
21860CPR
21870CPR      FILES USED --
21880CPR
21890CPR
21900CPR      LIBRARIES ACCESSED --
21910CPR      NONE
21920CPR
21930CPR
21940CPR      LOCAL VARIABLES --
21950CPR
21960CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
21970CPR
21980CPR
21990CPR      SPECIAL REMARKS/INSTRUCTIONS --
22000CPR          HOST = H6000
22010CPR
22020CPR
22030CPR=====
22240      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=21

```

22260C=====
22280      SUBROUTINE CORNCC
22300C=====
22320CPR
22330CPR=====
22340CPR
22350CPR
22360CPR      ROUTINE--          CORNCC
22370CPR
22380CPR      DATE/WRITTEN BY-- 11/10/82      RJ MAFFEO
22390CPR      DATE/REVISED BY--
22400CPR
22410CPR      FUNCTION/PURPOSE--
22420CPR
22430CPR      THIS SUB CALLS ALL ROUTINES NEEDED TO COMPUTE
22440CPR      AND STORE THE INFORMATION ASSOCIATED WITH THE
22450CPR      CORNER NODE WEIGHTING COEFFICIENTS
22460CPR
22470CPR
22480CPR
22490CPR      CALLING ARGUMENTS --
22500CPR      NAME      ATTRIBUTES      DEFINITION
22510CPR      NONE
22520CPR
22530CPR      COMMONS USED --
22540CPR
22550CPR      AFIL
22560CPR      ELMDAT
22570CPR      SIZE
22580CPR      WORK
22590CPR      CNTLGM
22600CPR
22610CPR      FUNCTIONS/ROUTINES CALLED --
22620CPR
22630CPR      RWELMS
22640CPR      SORT
22650CPR      CCNVEC
22660CPR      GETCRD
22670CPR      CINV
22680CPR      PUTWFI
22690CPR      RWCCF
22700CPR
22710CPR      FILES USED --
22720CPR
22730CPR      IERF      S
22740CPR
22750CPR      LIBRARIES ACCESSED --
22760CPR      NONE
22770CPR
22780CPR
22790CPR      LOCAL VARIABLES --
22800CPR
22810CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
22820CPR
22830CPR
22840CPR      SPECIAL REMARKS/INSTRUCTIONS --
22850CPR      HOST = H6000
22860CPR
22870CPR
22880CPR=====
24890      END

```

```

24910C=====
24930      SUBROUTINE CCNVEC(NOD,IRW,NLOC,NUMA,MCON,MXNCON,NSIZW)
24950C=====
24970CPR
24980CPR=====
24990CPR
25000CPR
25010CPR      ROUTINE--          CCNVEC
25020CPR
25030CPR      DATE/WRITTEN BY-- 11/10/82      RJ MAFFEO
25040CPR      DATE/REVISED BY--
25050CPR
25060CPR      FUNCTION/PURPOSE--
25070CPR
25080CPR          THIS SUB WRITES TO (CREATES) AND READS FROM THE
25090CPR          NODE -CONNECTION VECTOR (MCON)
25100CPR
25110CPR
25120CPR      CALLING ARGUMENTS --
25130CPR          NAME          ATTRIBUTES      DEFINITION
25140CPR
25150CPR          NOD            (I)              NODE NUMBER
25160CPR          IRW            (I)              READ/WRITE SWITCH(O-READ,1-WRITE)
25170CPR          NLOC            (I/O)           CURRENT CONNECTION TO NODE
25180CPR          NUMA            (O)              MAX # CONNECTIONS TO NODE
25190CPR          MCON            (I/O)           PACKED CONNECTION ARRAY
25200CPR          MXNCON         (O)              MAX # OF CONNECTIONS TO ANY NODE
25210CPR          NSIZW          (I)              SIXE OF WORK ARRAY (MCON HERE)
25220CPR
25230CPR      COMMONS USED --
25240CPR          AFIL
25250CPR
25260CPR
25270CPR      FUNCTIONS/ROUTINES CALLED --
25280CPR
25290CPR
25300CPR      FILES USED --
25310CPR
25320CPR          IERF      S
25330CPR
25340CPR      LIBRARIES ACCESSED --
25350CPR      NONE
25360CPR
25370CPR
25380CPR      LOCAL VARIABLES --
25390CPR
25400CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
25410CPR          THE NODE-CONNECTION VECTOR (MCON) IS KEYED
25420CPR          TO THE NODE NAME AND IS COMPOSED OF THE MAX
25430CPR          CONNECTION TO THAT NODE AND THE CURRENT CONNECTION TO
25440CPR          THAT NODE IN THE FORM --
25450CPR
25460CPR          MCON(NOD)=1000*MAX CONNECTION + CURRENT CONNECTION
25470CPR
25480CPR          THE INITIAL CURRENT COUNT IS EQUAL TO THE MAX CONNECTION.
25490CPR          AS THE CONNECTIONS ARE PROCESSED THE CURRENT COUNT IS
25500CPR          REDUCED BY ONE
25510CPR
25520CPR      SPECIAL REMARKS/INSTRUCTIONS --
25530CPR          HOST = H6000
25540CPR
25550CPR
25560CPR=====
26040      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=23

```

26060C=====
26080      SUBROUTINE GETCRD(NE,CRD,BRD,XCN, YCN,ZCN,JREC)
26100C=====
26120CPR
26130CPR=====
26140CPR
26150CPR
26160CPR      ROUTINE--          GETCRD
26170CPR
26180CPR      DATE/WRITTEN BY-- 11/11/82      RJ MAFFEO
26190CPR      DATE/REVISED BY--
26200CPR
26210CPR      FUNCTION/PURPOSE--
26220CPR
26230CPR      THIS SUB EXTRACTS THE APPROPRIATE SUBSET OF
26240CPR      COORDINATES FOR EACH CORNER NODE FORM THE
26250CPR      FULL SET OF COORDS OF EACH ELM
26260CPR
26270CPR
26280CPR
26290CPR
26300CPR
26310CPR      CALLING ARGUMENTS --
26320CPR      NAME          ATTRIBUTES          DEFINITION
26330CPR
26340CPR      NE          (I)          ELMEMENT NAME
26350CPR      CRD          (I)          FULL SET OF COORDS
26360CPR      BRD          (O)          SUBSET OF COORDS
26370CPR      XCN          (O)  >>>
26380CPR      YCN          (O)  >>>>>>>> COORDS OF CORNER
26390CPR      ZCN          (O)  >>>
26400CPR      JREC          (I)          CORNER LABEL
26410CPR
26420CPR      COMMONS USED --
26430CPR
26440CPR      AFIL
26450CPR
26460CPR      FUNCTIONS/ROUTINES CALLED --
26470CPR
26480CPR
26490CPR      FILES USED --
26500CPR
26510CPR      IERF      S
26520CPR
26530CPR      LIBRARIES ACCESSED --
26540CPR      NONE
26550CPR
26560CPR
26570CPR      LOCAL VARIABLES --
26580CPR
26590CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
26600CPR
26610CPR      THE ARRAY IDX IS USED TO DEFINE THE RELATIONSHIP
26620CPR      BETWEEN THE CORNER POINT AND THE APPROPRIATE 3 FACES
26630CPR      FOR EXAMPLE CORNER POINT 1 IS CONNECTED TO FACES
26640CPR      1,4,5
26650CPR      THEREFORE IDX(1,1)=1,IDX(1,2)=4,IDX(1,3)=5
26660CPR
26670CPR
26680CPR
26690CPR      SPECIAL REMARKS/INSTRUCTIONS --
26700CPR      HOST = H6000
26710CPR
26720CPR
26730CPR=====
27340      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=24

```

2736OC=====
2738O      SUBROUTINE PUTWFI(NOD,NE,NUMA,JREC,XSF,NPOS,ACWF)
2740OC=====
2742OCPR
2743OCPR=====
2744OCPR
2745OCPR
2746OCPR      ROUTINE--          PUTWFI
2747OCPR
2748OCPR      DATE/WRITTEN BY-- 11/11/82      RJ MAFFEO
2749OCPR      DATE/REVISED BY--
2750OCPR
2751OCPR      FUNCTION/PURPOSE--
2752OCPR
2753OCPR      THIS SUB POSITIONS THE WEIGHTING COEFF
2754OCPR      INFO INTO CORRECT LOCATIONS OF THE ACWF ARRAY
2755OCPR
2756OCPR
2757OCPR
2758OCPR
2759OCPR
2760OCPR      CALLING ARGUMENTS --
2761OCPR          NAME          ATTRIBUTES      DEFINITION
2762OCPR
2763OCPR          NOD            (I)             CORNER NODE NUMBER
2764OCPR          NE            (I)             ELM NUMBER
2765OCPR          NUMA           (I)             MAX NUMBER OF CONNECTIONS
2766OCPR          JREC           (I)             PACKED CORNER - FACE #
2767OCPR          XSFP           (I)             WEIGHTING COEFFS
2768OCPR          NPOS           (I)             STARTING POSITION FOR INFO
2769OCPR          ACWF           (O)             ARRAY CONTAINING INFO
2770OCPR
2771OCPR      COMMONS USED --
2772OCPR
2773OCPR          AFIL
2774OCPR
2775OCPR      FUNCTIONS/ROUTINES CALLED --
2776OCPR
2777OCPR
2778OCPR      FILES USED --
2779OCPR
2780OCPR
2781OCPR      LIBRARIES ACCESSED --
2782OCPR      NONE
2783OCPR
2784OCPR
2785OCPR      LOCAL VARIABLES --
2786OCPR
2787OCPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
2788OCPR
2789OCPR
2790OCPR      SPECIAL REMARKS/INSTRUCTIONS --
2791OCPR          HOST = H6000
2792OCPR
2793OCPR
2794OCPR=====
2826O      END

```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=25

```

28280C=====
28300      SUBROUTINE RWCCF(NOD,ACWF,NPOS,IRW)
28320C=====
28340CPR
28350CPR=====
28360CPR
28370CPR
28380CPR      ROUTINE--          RWCCF
28390CPR
28400CPR      DATE/Written BY-- 11/11/82      RJ MAFFED
28410CPR      DATE/REVISED BY--
28420CPR
28430CPR      FUNCTION/PURPOSE--
28440CPR
28450CPR      THIS ROUTINE READS AND WRITES THE CORNER NODE
28460CPR      COEFF INFO TO AND FROM THE DATA BASE
28470CPR
28480CPR
28490CPR
28500CPR
28510CPR
28520CPR      CALLING ARGUMENTS --
28530CPR          NAME          ATTRIBUTES      DEFINITION
28540CPR
28550CPR          NOD            (I/O)            NODE NAME (RECORD #)
28560CPR          ACWF          (I/O)            ARRAY OF INFO
28570CPR          NPOS          (I)              # OF TERMS TO BE READ/WRITTEN
28580CPR          IRW           (I)              READ/WRITE SWITCH(O-READ,1-WRITE)
28590CPR
28600CPR      COMMONS USED --
28610CPR
28620CPR          AFIL
28630CPR
28640CPR      FUNCTIONS/ROUTINES CALLED --
28650CPR
28660CPR
28670CPR      FILES USED --
28680CPR
28690CPR          IRCCF      R
28700CPR
28710CPR      LIBRARIES ACCESSED --
28720CPR      NONE
28730CPR
28740CPR
28750CPR      LOCAL VARIABLES --
28760CPR
28770CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
28780CPR
28790CPR
28800CPR      SPECIAL REMARKS/INSTRUCTIONS --
28810CPR          HOST = H6000
28820CPR
28830CPR
28840CPR=====
29120      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=26

```

29140C=====
29160      SUBROUTINE GETMPE(MP,NWDMPE)
29180C=====
29200CPR
29210CPR=====
29220CPR
29230CPR
29240CPR      ROUTINE--          GETMPE
29250CPR
29260CPR      DATE/WRITTEN BY-- 09/19/83      RJ MAFFEO
29270CPR      DATE/REVISED BY--
29280CPR
29290CPR      FUNCTION/PURPOSE--
29300CPR
29310CPR      THIS ROUTINE READS THE FIRST TIME STEP TEMPERATURE
29320CPR      SOLUTION AND EXTRACTS THE ELM NAME AND RELATIVE POSITION
29330CPR      OF THIS ELM IN THE OUTPUT LIST.IT THEN CALLS MPERW TO
29340CPR      STORE THIS INFO INTO THE MP ARRAY
29350CPR
29360CPR
29370CPR
29380CPR
29390CPR
29400CPR      CALLING ARGUMENTS --
29410CPR          NAME          ATTRIBUTES          DEFINITION
29420CPR
29430CPR          MP              (I/O)              ELM MATRIX POSITION ARRAY
29440CPR          NWDMPE          (I)                # OF WORDS IN MP ARRAY
29450CPR
29460CPR      COMMONS USED --
29470CPR
29480CPR          AFIL
29490CPR          SIZE
29500CPR          CNTLFL
29510CPR
29520CPR      FUNCTIONS/ROUTINES CALLED --
29530CPR
29540CPR          TFLSET
29550CPR          HTFIO
29560CPR          MPERW
29570CPR
29580CPR      FILES USED --
29590CPR
29600CPR
29610CPR      LIBRARIES ACCESSED --
29620CPR      NONE
29630CPR
29640CPR
29650CPR      LOCAL VARIABLES --
29660CPR
29670CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
29680CPR
29690CPR
29700CPR      SPECIAL REMARKS/INSTRUCTIONS --
29710CPR          HOST = H6000
29720CPR
29730CPR
29740CPR=====
30240      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=27

```

30260C=====
30280      SUBROUTINE MPERW(MP,NE,NPOS,IRW)
30300C=====
30320CPR
30330CPR=====
30340CPR
30350CPR
30360CPR      ROUTINE--          MPERW
30370CPR
30380CPR      DATE/WRITTEN BY-- 10/08/83      RJ MAFFED
30390CPR      DATE/REVISED BY--
30400CPR
30410CPR      FUNCTION/PURPOSE--
30420CPR
30430CPR      THIS ROUTINE PACKS AND UNPACKS THE INFO NEEDED TO RELATE
30440CPR      THE ELEMENT NAME TO THE RELATIVE POSITION OF THE ELEMENT
30450CPR      IN THE OUTPUT LIST
30460CPR
30470CPR
30480CPR
30490CPR
30500CPR      CALLING ARGUMENTS --
30510CPR          NAME          ATTRIBUTES      DEFINITION
30520CPR
30530CPR          MP            (I/O)            MATRIX POSITION ARRAY
30540CPR          NE            (I)              ELM NAME
30550CPR          NPOS          (I/O)            RELATIVE POSITION IN OUTPUT
30560CPR          IRW            (I)              SWITCH(0--READ,1--WRITE)
30570CPR
30580CPR      COMMONS USED --  NONE
30590CPR
30600CPR
30610CPR      FUNCTIONS/ROUTINES CALLED --
30620CPR          MOD (FORTRAN SUPPLIED)
30630CPR
30640CPR      FILES USED --
30650CPR
30660CPR          06          S
30670CPR
30680CPR      LIBRARIES ACCESSED --
30690CPR      NONE
30700CPR
30710CPR
30720CPR      LOCAL VARIABLES --
30730CPR
30740CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
30750CPR
30760CPR      THIS ROUTINE USES DECIMAL PACKING TO PACK 2 POSITIONS
30770CPR      PER WORD OF THE MP ARRAY IT MAKES USE OF THE FORTRAN
30780CPR      SUPPLIED 'MOD' FUNCTION
30790CPR      THE POSITIONS ARE PACKED AS FOLLOWS --
30800CPR          POSITION 1 >>> LEFT HAND SIDE OF WORD
30810CPR          POSITION 2 >>> RIGHT HAND SIDE OF WORD
30820CPR      THE MAX POSITION THAT CAN BE PACKED IS NPOS=9999
30830CPR      THE ADVANTAGE OF THIS SCHEME IS THAT THE DIMENSION OF
30840CPR      MP NEED ONLY BE 4000 TO STORE ELMS UP TO 8000
30850CPR
30860CPR      FOR EXAMPLE
30870CPR          IF THE POSITION OF ELM 1 IS 987
30880CPR          AND THE POSITION OF ELM 2 IS 79
30890CPR          THEN THE VALUE STORED IN MP(1) IS 09870079
30900CPR          NOTE THAT MP(1) CONTAINS THE POSITIONS OF BOTH
30910CPR          ELMS 1 AND 2
30920CPR
30930CPR      SPECIAL REMARKS/INSTRUCTIONS --
30940CPR          HOST = H6000
30950CPR

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=28

30960CPR

30970CPR=====

31620 END

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=29

```
31640C=====
31660      SUBROUTINE SETFLP(NNEW,NOLD)
31680C=====
31700CPR
31710CPR=====
31720CPR
31730CPR
31740CPR      ROUTINE--          SETFLP
31750CPR
31760CPR      DATE/WRITTEN BY-- 09/19/83      RJ MAFFEO
31770CPR      DATE/REVISED BY--
31780CPR
31790CPR      FUNCTION/PURPOSE--
31800CPR
31810CPR      THIS ROUTINE SETS THE H T  INPUT FILE TO THE CORRECT
31820CPR      POSITION TO READ THE SOLUTION DESIRED
31830CPR
31840CPR
31850CPR
31860CPR
31870CPR
31880CPR      CALLING ARGUMENTS --
31890CPR          NAME          ATTRIBUTES      DEFINITION
31900CPR
31910CPR          NNEW          (I)              FILE DESIRED
31920CPR          NOLD          (I)              CURRENT FILE COUNT
31930CPR                                         OF THT OUTPUT FILE
31940CPR
31950CPR      COMMONS USED --
31960CPR
31970CPR          AFIL
31980CPR
31990CPR      FUNCTIONS/ROUTINES CALLED --
32000CPR
32010CPR          TFLSET
32020CPR          HTFPAS
32030CPR
32040CPR      FILES USED --
32050CPR
32060CPR
32070CPR      LIBRARIES ACCESSED --
32080CPR      NONE
32090CPR
32100CPR
32110CPR      LOCAL VARIABLES --
32120CPR
32130CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
32140CPR
32150CPR
32160CPR      SPECIAL REMARKS/INSTRUCTIONS --
32170CPR          HOST = H6000
32180CPR
32190CPR
32200CPR=====
32550      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=30

```
3257OC=====
32590      SUBROUTINE RDTEMP(TEMP,NFLVAL)
3261OC=====
3263OCPR
3264OCPR=====
3265OCPR
3266OCPR
3267OCPR      ROUTINE--          RDTEMP
3268OCPR
3269OCPR      DATE/WRITTEN BY-- 09/19/83      RJ MAFFEO
3270OCPR      DATE/REVISED BY--
3271OCPR
3272OCPR      FUNCTION/PURPOSE--
3273OCPR
3274OCPR      THIS ROUTINE READS THE TEMPERATURE INFO FOR THE DESIRED
3275OCPR      TIME STEP FROM THE THT OUTPUT FILE AND STORES IT IN
3276OCPR      THE TEMP ARRAY
3277OCPR
3278OCPR
3279OCPR
3280OCPR
3281OCPR
3282OCPR      CALLING ARGUMENTS --
3283OCPR          NAME      ATTRIBUTES      DEFINITION
3284OCPR
3285OCPR          TEMP      (O)              TEMPER  STORAGE ARRAY
3286OCPR          NFLVAL    (I)              CURRENT  FILE COUNT
3287OCPR
3288OCPR      COMMONS USED --
3289OCPR
3290OCPR          AFIL
3291OCPR          SIZE
3292OCPR          CNTLFL
3293OCPR
3294OCPR      FUNCTIONS/ROUTINES CALLED --
3295OCPR
3296OCPR          HTFIO
3297OCPR
3298OCPR      FILES USED --
3299OCPR
3300OCPR          IERF      S
3301OCPR
3302OCPR      LIBRARIES ACCESSED --
3303OCPR      NONE
3304OCPR
3305OCPR
3306OCPR      LOCAL VARIABLES --
3307OCPR
3308OCPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
3309OCPR
3310OCPR
3311OCPR      SPECIAL REMARKS/INSTRUCTIONS --
3312OCPR          HOST = H6000
3313OCPR
3314OCPR
3315OCPR=====
3367O      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=31

```

3369OC=====
3371O      SUBROUTINE CWCRT(MP,TEMP,TIMVAL,NNODMX,NNODMN,TCRIT,MXNCON)
3373OC=====
3375OCPR
3376OCPR=====
3377OCPR
3378OCPR
3379OCPR      ROUTINE--          CWCRT
3380OCPR
3381OCPR      DATE/WRITTEN BY-- 09/19/83      RJ MAFFEO
3382OCPR      DATE/REVISED BY--
3383OCPR
3384OCPR      FUNCTION/PURPOSE--
3385OCPR
3386OCPR      THIS ROUTINE COMPUTES A UNIQUE TEMPERATURE FOR EACH
3387OCPR      CORNER OF THE HEAT TRANSFER ELM BASED ON THE INDIVIDUAL
3388OCPR      TEMPERATURES FOR EACH ELM CONNECTION
3389OCPR
3390OCPR
3391OCPR
3392OCPR
3393OCPR
3394OCPR      CALLING ARGUMENTS --
3395OCPR      NAME          ATTRIBUTES      DEFINITION
3396OCPR
3397OCPR      MP          (I)          ELM MATRIX POSITION ARRAY
3398OCPR      TEMP        (I)          ELM TEMPERATURE ARRAY
3399OCPR      TIMVAL      (I)          CURRENT VALUE OF TIME
3400OCPR      NNODMX     (I)          MAX NODE NAME
3401OCPR      NNODMN     (I)          MIN NODE NAME
3402OCPR      TCRIT      (I)          TEMPERATURE USED IN MAXTDF TO
3403OCPR                      DETERMINE IF CORNER TEMPERATURE
3404OCPR                      DIFFERENCE SHOULD BE OUTPUT
3405OCPR      MXNCON     (I)          MAX NUMBER CONNECTIONS TO
3406OCPR                      ANY NODE
3407OCPR
3408OCPR      COMMONS USED --
3409OCPR
3410OCPR      AFIL
3411OCPR
3412OCPR      FUNCTIONS/ROUTINES CALLED --
3413OCPR
3414OCPR      RWCCF
3415OCPR      UNPAKF
3416OCPR      MPERW
3417OCPR      FCRNTP
3418OCPR      MAXTDF
3419OCPR      WRITMP
3420OCPR      FILREW
3421OCPR
3422OCPR      FILES USED --
3423OCPR
3424OCPR
3425OCPR      LIBRARIES ACCESSED --
3426OCPR      NONE
3427OCPR
3428OCPR
3429OCPR      LOCAL VARIABLES --
3430OCPR
3431OCPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
3432OCPR
3433OCPR
3434OCPR      SPECIAL REMARKS/INSTRUCTIONS --
3435OCPR      HOST = H6000
3436OCPR
3437OCPR
3438OCPR=====

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=32

35450

END

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=33

```

35470C=====
35490      SUBROUTINE UNPAKF(NPACK,NAME,IFACE)
35510C=====
35530CPR
35540CPR=====
35550CPR
35560CPR
35570CPR      ROUTINE--          UNPAKF
35580CPR
35590CPR      DATE/WRITTEN BY-- 11/29/82      RJ MAFFED
35600CPR      DATE/REVISED BY--
35610CPR
35620CPR      FUNCTION/PURPOSE--
35630CPR
35640CPR      THIS ROUTINE UNPACKS THE CORNER-FACE NUMBER INTO AN CORNER
35650CPR      NUMBER(NAME) AND THE FACE ARRAY(IFACE)
35660CPR
35670CPR
35680CPR
35690CPR
35700CPR
35710CPR      CALLING ARGUMENTS --
35720CPR          NAME          ATTRIBUTES      DEFINITION
35730CPR
35740CPR          NPACK          (I)              PACKED CORNER-FACE #
35750CPR          NAME          (O)              CORNER NAME
35760CPR          IFACE          (O)              FACE INFO ARRAY
35770CPR
35780CPR      COMMONS USED --
35790CPR
35800CPR          AFIL
35810CPR
35820CPR      FUNCTIONS/ROUTINES CALLED --
35830CPR
35840CPR
35850CPR      FILES USED --
35860CPR
35870CPR          IERF          S
35880CPR
35890CPR      LIBRARIES ACCESSED --
35900CPR      NONE
35910CPR
35920CPR
35930CPR      LOCAL VARIABLES --
35940CPR
35950CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
35960CPR
35970CPR      NOTE  PACK CORNER-FACE # = CORNER NAME*1000000+IFACPC
35980CPR          WHERE IFACPC IS A SINGLE INTEGER THAT CONTAINS ALL
35990CPR          ALL THE FACE NUMBERS
36000CPR      FOR EXAMPLE
36010CPR          IF THE FACE NUMBERS WERE 6,5,4,3,2,1
36020CPR          THEN IFACPC=654321
36030CPR
36040CPR
36050CPR
36060CPR      SPECIAL REMARKS/INSTRUCTIONS --
36070CPR          HOST = H6000
36080CPR
36090CPR
36100CPR=====
35480      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=34

```

36500C=====
36520      SUBROUTINE FCRNTP(TEMP,MPOS,IFACE,NCORN,CTP,COF)
36540C=====
36560CPR
36570CPR=====
36580CPR
36590CPR
36600CPR      ROUTINE--          FCRNTP
36610CPR
36620CPR      DATE/WRITTEN BY-- 11/29/82      RJ MAFFEO
36630CPR      DATE/REVISED BY--
36640CPR
36650CPR      FUNCTION/PURPOSE--
36660CPR
36670CPR      THIS ROUTINE EXTRACTS THE 7 TEMPERATURES ASSC WITH A
36680CPR      CONNECTING ELM IT THEN FINDS THE 4 TEMPS ASSC WITH
36690CPR      THE CORNER OF INTEREST AND DOES THE MULTIPLICATION
36700CPR      TO COMPUTE THE TEMPERATURE FOR THIS CORNER BASED ON
36710CPR      THIS ELM CONNECTION
36720CPR
36730CPR
36740CPR
36750CPR
36760CPR
36770CPR      CALLING ARGUMENTS --
36780CPR          NAME          ATTRIBUTES      DEFINITION
36790CPR
36800CPR          TEMP          (I)              ELM TEMPER  ARRAY
36810CPR          MPOS          (I)              MATRIX POS  OF THE ELM
36820CPR          IFACE          (I)              ELM FACE  ARRAY
36830CPR          NCORN          (I)              CORNER  NUMBER
36840CPR          CTP            (O)              CORNER  TEMPERATURE
36850CPR          COF          (I)              CORNER  COEFF  ARRAY
36860CPR
36870CPR      COMMONS USED --
36880CPR
36890CPR          AFIL
36900CPR
36910CPR      FUNCTIONS/ROUTINES CALLED --
36920CPR
36930CPR
36940CPR      FILES USED --
36950CPR
36960CPR
36970CPR      LIBRARIES ACCESSED --
36980CPR      NONE
36990CPR
37000CPR
37010CPR      LOCAL VARIABLES --
37020CPR
37030CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
37040CPR
37050CPR
37060CPR      SPECIAL REMARKS/INSTRUCTIONS --
37070CPR          HOST = H6000
37080CPR
37090CPR
37100CPR=====
37970      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=35

```

37990C=====
38010      SUBROUTINE MAXTDF(NEL,CRNT,NCON,NDONE,TIMVAL,TCRIT,IREC)
38030C=====
38050CPR
38060CPR=====
38070CPR
38080CPR
38090CPR      ROUTINE--          MAXTDF
38100CPR
38110CPR      DATE/WRITTEN BY-- 11/30/82      RJ MAFFEO
38120CPR      DATE/REVISED BY--
38130CPR
38140CPR      FUNCTION/PURPOSE--
38150CPR
38160CPR      THIS ROUTINE COMPUTES THE MAX DIFFERENCE BETWEEN
38170CPR      ALL THE POSSIBLE CORNER TEMPERATURES AT A CORNER
38180CPR      AND FLAGS IT IF IT IS GREATER THAN TCRIT
38190CPR
38200CPR
38210CPR
38220CPR
38230CPR
38240CPR      CALLING ARGUMENTS --
38250CPR          NAME          ATTRIBUTES      DEFINITION
38260CPR
38270CPR          NEL            (I)              ARRAY OF ELMS CONNECTED TO THIS NODE
38280CPR          CRNT           (I)              ARRAY OF TEMPS FOR THIS NODE
38290CPR          NCON            (I)              NUMBER OF ELM CONNECTED TO THIS NODE
38300CPR          NDONE          (I)              PAGE HEADER COUNTER
38310CPR                                     (=0 WRITE HEADER, NOT = 0 DONT WRITE)
38320CPR          TIMVAL         (I)              CURRENT VALUE OF TIME
38330CPR          TCRIT          (I)              CRITICAL TEMP FOR PRINTOUT
38340CPR                                     IF MAX DIFF < TCRIT -- DONT WRITE
38350CPR                                     IF MAX DIFF > TCRIT -- WRITE
38360CPR          IREC           (I)              RECORD NUMBER(NODE NUMBER)OF CORNER
38370CPR
38380CPR      COMMONS USED --
38390CPR
38400CPR          AFIL
38410CPR
38420CPR      FUNCTIONS/ROUTINES CALLED --
38430CPR
38440CPR          SORT
38450CPR
38460CPR      FILES USED --
38470CPR
38480CPR          INOT6      S
38490CPR
38500CPR      LIBRARIES ACCESSED --
38510CPR          NONE
38520CPR
38530CPR
38540CPR      LOCAL VARIABLES --
38550CPR
38560CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
38570CPR
38580CPR
38590CPR      SPECIAL REMARKS/INSTRUCTIONS --
38600CPR          HOST = H6000
38610CPR
38620CPR
38630CPR=====
39160      END

```

```

39180C=====
39200      SUBROUTINE WRITMP(TBAR,TIMVAL,NOD,NDONE)
39220C=====
39240CPR
39250CPR=====
39260CPR
39270CPR
39280CPR      ROUTINE--          WRITMP
39290CPR
39300CPR      DATE/WRITTEN BY-- 11/30/82      RJ MAFFEO
39310CPR      DATE/REVISED BY--
39320CPR
39330CPR      FUNCTION/PURPOSE--
39340CPR
39350CPR      THIS ROUTINE WRITES THE VALUES OF THE TEMPERATURES AT THE
39360CPR      STRESS POINTS TO HARD COPY AND TO THE OUTPUT FILE
39370CPR
39380CPR
39390CPR
39400CPR
39410CPR
39420CPR      CALLING ARGUMENTS --
39430CPR          NAME          ATTRIBUTES      DEFINITION
39440CPR
39450CPR          TBAR           (I)              TEMPERATURE AT THE STRESS POINT
39460CPR          TIMVAL        (I)              CURRENT VALUE OF TIME
39470CPR          NOD           (I)              STRESS POINT NAME(NODE #)
39480CPR          NDONE         (I)              SWITCH
39490CPR                                     =0      -- WRITER PAGE HEADER
39500CPR                                     NOT 0  -- DO NOT WRITE HEADER
39510CPR                                     --1      -- FLUSH BUFFERS
39520CPR
39530CPR      COMMONS USED --
39540CPR
39550CPR          AFIL
39560CPR
39570CPR      FUNCTIONS/ROUTINES CALLED --
39580CPR
39590CPR
39600CPR      FILES USED --
39610CPR
39620CPR          INOT6      S
39630CPR          IOUTF      S
39640CPR
39650CPR      LIBRARIES ACCESSED --
39660CPR      NONE
39670CPR
39680CPR
39690CPR      LOCAL VARIABLES --
39700CPR
39710CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
39720CPR
39730CPR
39740CPR      SPECIAL REMARKS/INSTRUCTIONS --
39750CPR          HOST = H6000
39760CPR
39770CPR
39780CPR=====
40580      END

```

```

40600C=====
40620      SUBROUTINE CINV(CD,XSP,YSP,ZSP,XSF,NE,JCN)
40640C=====
40660CPR
40670CPR=====
40680CPR
40690CPR
40700CPR ROUTINE--   CINV
40710CPR
40720CPR DATE/WRITTEN BY--  9/10/82   RJ MAFFEO
40730CPR DATE/REVISED BY-- 01/12/83  RJ MAFFEO
40740CPR
40750CPR FUNCTION/PURPOSE--
40760CPR THIS SUB FIND THE VALUES OF THE SHAPE FUNCTIONS FOR
40770CPR ANY POINT NEAR A TETRAHEDRAL ELEMENT THE SHAPE FUNCTIONS
40780CPR ARE BASED ON THE SIMPLEX TETRAHEDRAL ELEMENT
40790CPR IT FIRST FINDS THE INVERSE OF THE COORDINATE MATRIX (-C-)
40800CPR IT THEN THENS MULTIPLIES THIS INVERSE INTO THE COORD
40810CPR VECTOR (1 X Y Z) TO FIND THE SHAPE FUNCTIONS
40820CPR THIS ROUTINE CAN BE USED TO COMPUTE THE WEIGHTING
40830CPR COEFF FOR THE CORNERS OF A THTD ELM BASED ON THE TEMPERATURES
40840CPR AT THE CENTER OF THE FACES AND THE CENTROID
40850CPR
40860CPR CALLING ARGUMENTS--
40870CPR      NAME      ATTRIBUTES      DEFINITION
40880CPR      CD        (I) 12X1 ARRAY  COORDINATE ARRAY
40890CPR                                     X,Y,Z COORDS OF CORNERS OF TETRA
40900CPR      XSP      (I)
40910CPR      YSP      (I)
40920CPR      ZSP      (I)
40930CPR                                     COORDS OF THE POINT FOR WHICH
40940CPR      XSF      (O) 5X1 ARRAY  THE SHAPE FUNCTIONS ARE DESIRED
40950CPR      NE       (I)            VECTOR OF SHAPE FUNCTIONS
40960CPR      JCN      (I)            ELMEMENT NAME
40970CPR                                     CORNER NUMBER
40980CPR
40990CPR
41000CPR FILES USED--
41010CPR      AFIL
41020CPR
41030CPR COMMONS USED--
41040CPR      NONE
41050CPR
41060CPR
41070CPR ROUTINES CALLED --
41080CPR      NONE
41090CPR
41100CPR LIBRARIES ACCESSED--
41110CPR      NONE
41120CPR
41130CPR LOCAL VARIABLES--
41140CPR      INX      18X1      THIS VECTOR IS USED TO SELECT THE
41150CPR                                     CORRECT INDICES OF THE C MATRIX FOR
41160CPR                                     EACH ELEMENT OF THE C INVERSE MATRIX
41170CPR      CI       4X4      INVERSE OF THE COORD MATRIX(-C-)
41180CPR      C       5X4      2D MATRIX CONTAINING CORRDS OF TETRA
41190CPR
41200CPR
41210CPR SPECIAL COMMENTS ABOUT THIS ROUTINE --
41220CPR
41230CPR      NOTE THAT THE COORD MATRIX(-C-) IS AUGMENTED WITH
41240CPR      A ROW OF ONES THIS ROW OF ONES IS USED IN THE COMPUTATION
41250CPR      OF THE INVERSE
41260CPR
41270CPR      NOTE XSF(5)=1/DIST
41280CPR      DIST=DISTANCE FROM CENTROID TO CORNER
41290CPR

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=38

41300CPR NOTE A LOCAL COORD SYSTEM IS ESTABLISHED FOR EACH ELM
41310CPR THIS SYSTEM IS FORMED AS Q(I)-QC
41320CPR WHERE Q(I) IS THE COORDS OF THE VERTICES
41330CPR AND QC IS THE COORDS OF THE CENTROID OF THE ELM
41340CPR
41350CPR
41360CPR
41370CPR SPECIAL REMARKS/INSTRUCTIONS--
41380CPR HOST = H6000
41390CPR
41400CPR*****
42720 END

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=39

42740CC=====

42760 SUBROUTINE PRNTIM(ISTOP,I1,TB,TE)

42930 END

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=40

```

42950C=====
42970      SUBROUTINE EMINMX(XC,YC,ZC,CADD,XYZV,NEMNMX,NE,NNPE)
42990C=====
43010CPR
43020CPR=====
43030CPR
43040CPR
43050CPR      ROUTINE--          EMINMX
43060CPR
43070CPR      DATE/WRITTEN BY-- 04/05/83      RJ MAFFEO
43080CPR      DATE/REVISED BY--
43090CPR
43100CPR      FUNCTION/PURPOSE--
43110CPR
43120CPR      THIS ROUTINE FINDS THE MIN AND MAX XYZ COORDS
43130CPR      OF THE HEAT TRANSFER ELEMENT AND STORES THIS INFO
43140CPR      INTO THE XYZV ARRAY
43150CPR      IT ALSO KEEPS TRACK OF THE ELMS WITH THE SMALLEST
43160CPR      AND LARGEST XYZ COORDS AND STORES THEM IN THE
43170CPR      NEMNMX ARRAY
43180CPR
43190CPR
43200CPR      CALLING ARGUMENTS --
43210CPR          NAME          ATTRIBUTES      DEFINITION
43220CPR
43230CPR          XC          (I) >>
43240CPR          YC          (I) >>>>  COORD ARRAYS FOR HT ELM VERTICES
43250CPR          ZC          (I) >>
43260CPR          CADD          (I)          CONSTANT COORD ADDER
43290CPR          NE          (I)          CURRENT ELM NAME
43300CPR          NNPE          (I)          # OF NODES PER ELM
43310CPR
43320CPR      COMMONS USED --
43330CPR
43340CPR
43350CPR      FUNCTIONS/ROUTINES CALLED --
43360CPR
43370CPR          SORT
43380CPR
43390CPR      FILES USED --
43400CPR
43410CPR
43420CPR      LIBRARIES ACCESSED --
43430CPR      NONE
43440CPR
43450CPR
43460CPR      LOCAL VARIABLES --
43470CPR
43480CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
43490CPR
43500CPR
43510CPR      SPECIAL REMARKS/INSTRUCTIONS --
43520CPR          HOST = H6000
43530CPR
43540CPR
43550CPR=====
43910      END

```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=41

```

43930C=====
43950      SUBROUTINE FTIME(A1)
43970C=====
43990CPR
44000CPR=====
44010CPR
44020CPR
44030CPR      ROUTINE--          FTIME
44040CPR
44050CPR      DATE/WRITTEN BY-- 04/05/83      RJ MAFFED
44060CPR      DATE/REVISED BY--
44070CPR
44080CPR      FUNCTION/PURPOSE--
44090CPR
44100CPR      THIS ROUTINE CALLS THE APPROPRIATE SYSTEM ROUTINE
44110CPR      TO MEASURE THE PROCESSOR TIME
44120CPR
44130CPR
44140CPR      CALLING ARGUMENTS --
44150CPR          NAME          ATTRIBUTES      DEFINITION
44160CPR
44170CPR          A1              (0)          PROCESSOR TIME
44180CPR
44190CPR      COMMONS USED --      NONE
44200CPR
44210CPR
44220CPR      FUNCTIONS/ROUTINES CALLED --
44230CPR
44240CPR          PTIME -- HONEY WELL TIMER
44250CPR
44260CPR      FILES USED --
44270CPR
44280CPR
44290CPR      LIBRARIES ACCESSED --
44300CPR      NONE
44310CPR
44320CPR
44330CPR      LOCAL VARIABLES --
44340CPR
44350CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
44360CPR
44370CPR
44380CPR      SPECIAL REMARKS/INSTRUCTIONS --
44390CPR          HOST = H6000
44400CPR
44410CPR
44420CPR=====
44470      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=42

```
44490C=====
44510      SUBROUTINE SSURCH
44530C=====
44550CPR
44560CPR=====
44570CPR
44580CPR
44590CPR      ROUTINE--          SSURCH
44600CPR
44610CPR      DATE/WRITTEN BY-- 04/05/83      RJ MAFFEO
44620CPR      DATE/REVISED BY--
44630CPR
44640CPR      FUNCTION/PURPOSE--
44650CPR
44660CPR      THIS ROUTINE IS THE MINI-EXEC FOR THE STRESS POINT
44670CPR      TO HEAT TRANSFER ELM SEARCH ROUTINES IT CALLS ALL
44680CPR      REQUIRED ROUTINES TO FIND WHICH HT ELM CONTAINS
44690CPR      EACH STRESS POINT AND THEN FINDS THE WEIGHTING COEFFS
44700CPR      FOR THAT STRESS POINT
44710CPR
44720CPR
44730CPR
44740CPR      CALLING ARGUMENTS --
44750CPR          NAME          ATTRIBUTES      DEFINITION
44760CPR
44770CPR
44780CPR      COMMONS USED --
44790CPR
44800CPR          AFIL
44810CPR          SIZE
44820CPR          SPDAT
44830CPR
44840CPR      FUNCTIONS/ROUTINES CALLED --
44850CPR
44860CPR          KNTSPF
44870CPR          RFILCR
44880CPR          ZERORF
44890CPR          GNLSUR
44900CPR
44910CPR      FILES USED --
44920CPR
44930CPR
44940CPR      LIBRARIES ACCESSED --
44950CPR      NONE
44960CPR
44970CPR
44980CPR      LOCAL VARIABLES --
44990CPR
45000CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
45010CPR
45020CPR
45030CPR      SPECIAL REMARKS/INSTRUCTIONS --
45040CPR          HOST = H6000
45050CPR
45060CPR
45070CPR=====
45600      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=43

```

45620C=====
45640      SUBROUTINE KNTSPF(NUMSP,MXNSP)
45660C=====
45680CPR
45690CPR=====
45700CPR
45710CPR
45720CPR      ROUTINE--          KNTSPF
45730CPR
45740CPR      DATE/WRITTEN BY-- 04/05/83      RJ MAFFEO
45750CPR      DATE/REVISED BY--
45760CPR
45770CPR      FUNCTION/PURPOSE--
45780CPR
45790CPR      THIS ROUTINE COUNTS THE NUMBER OF STRESS POINTS
45800CPR      TO BE INTERFACED FROM THE STRESS NODE FILE AND
45810CPR      ALSO FINDS THE LARGEST STRESS NODE NAME
45820CPR
45830CPR
45840CPR
45850CPR      CALLING ARGUMENTS --
45860CPR          NAME          ATTRIBUTES      DEFINITION
45870CPR
45880CPR          NUMSP          (0)              # OF STRESS NODES
45890CPR          MXNSP          (0)              LARGEST STRESS NODE NAME
45900CPR
45910CPR      COMMONS USED --
45920CPR
45930CPR          AFIL
45940CPR
45950CPR      FUNCTIONS/ROUTINES CALLED --
45960CPR
45970CPR          FILREW
45980CPR          RDSPF
45990CPR
46000CPR      FILES USED --
46010CPR
46020CPR          IERF          S
46030CPR
46040CPR      LIBRARIES ACCESSED --
46050CPR      NONE
46060CPR
46070CPR
46080CPR      LOCAL VARIABLES --
46090CPR
46100CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
46110CPR
46120CPR
46130CPR      SPECIAL REMARKS/INSTRUCTIONS --
46140CPR          HOST = H6000
46150CPR
46160CPR
46170CPR=====
46560      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=44

```
46580C=====
46600      SUBROUTINE RFILCR
46620C=====
46640CPR
46650CPR=====
46660CPR
46670CPR
46680CPR      ROUTINE--          RFILCR
46690CPR
46700CPR      DATE/WRITTEN BY-- 04/06/83      RJ MAFFED
46710CPR      DATE/REVISED BY--
46720CPR
46730CPR      FUNCTION/PURPOSE--
46740CPR
46750CPR      THIS ROUTINE SIZES ALL THE RANDOM FILES
46760CPR      IT CALLS THE APPROPRIATE SYSTEM ROUTINE TO DEFINE
46770CPR      THE WIDTH OF THE FILE
46780CPR
46790CPR
46800CPR
46810CPR      CALLING ARGUMENTS --
46820CPR          NAME          ATTRIBUTES      DEFINITION
46830CPR
46840CPR
46850CPR      COMMONS USED --
46860CPR
46870CPR
46880CPR      FUNCTIONS/ROUTINES CALLED --
46890CPR
46900CPR          RANSIZ  -- HONEYWELL ROUTINE TO SET WIDTH OF FILE
46910CPR          DEFINE FILE -- IBM ROUTINE TO SET WIDTH
46920CPR
46930CPR      FILES USED --
46940CPR
46950CPR
46960CPR      LIBRARIES ACCESSED --
46970CPR      NONE
46980CPR
46990CPR
47000CPR      LOCAL VARIABLES --
47010CPR
47020CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
47030CPR
47040CPR
47050CPR      SPECIAL REMARKS/INSTRUCTIONS --
47060CPR          HOST = H6000
47070CPR
47080CPR
47090CPR=====
47800      END
```

TITLE= DATE =07/15/85 TIME = 23 42 PAGE=45

```

47820C=====
47840      SUBROUTINE ZERORF(IFC,MXN,NWD,A)
47860C=====
47880CPR
47890CPR=====
47900CPR
47910CPR
47920CPR      ROUTINE--          ZERORF
47930CPR
47940CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFEO
47950CPR      DATE/REVISED BY--
47960CPR
47970CPR      FUNCTION/PURPOSE--
47980CPR
47990CPR      THIS ROUTINE INITIALIZES RANDOM FILES
48000CPR
48010CPR
48020CPR      CALLING ARGUMENTS --
48030CPR          NAME          ATTRIBUTES      DEFINITION
48040CPR
48050CPR          IFC          (I)          FILECODE
48060CPR          MXN          (I)          MAX RECORD NUMBER
48070CPR          NWD          (I)          # OF WORDS IN -A- ARRAY
48080CPR          A            (I)          INITIALIZEING ARRAY
48090CPR
48100CPR      COMMONS USED --
48110CPR
48120CPR
48130CPR      FUNCTIONS/ROUTINES CALLED --
48140CPR
48150CPR
48160CPR      FILES USED --
48170CPR
48180CPR          IFC          R
48190CPR
48200CPR      LIBRARIES ACCESSED --
48210CPR      NONE
48220CPR
48230CPR
48240CPR      LOCAL VARIABLES --
48250CPR
48260CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
48270CPR
48280CPR
48290CPR      SPECIAL REMARKS/INSTRUCTIONS --
48300CPR          HOST = H6000
48310CPR
48320CPR
48330CPR=====
48390      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=46

```
48410C=====
48430      SUBROUTINE GNLSUR
48450C=====
48470CPR
48480CPR=====
48490CPR
48500CPR
48510CPR      ROUTINE--          GNLSUR
48520CPR
48530CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFED
48540CPR      DATE/REVISED BY--
48550CPR
48560CPR      FUNCTION/PURPOSE--
48570CPR
48580CPR      THIS ROUTINE IS CALL BY THE SEARCH MINI-EXEC (SSURCH)
48590CPR      IT INITIATES THE SEARCH THRU THE HEAT TRANSFER ELMS
48600CPR      TO FIND WHICH ELM CONTAINS EACH STRESS POINT
48610CPR
48620CPR
48630CPR
48640CPR      CALLING ARGUMENTS --
48650CPR          NAME          ATTRIBUTES      DEFINITION
48660CPR
48670CPR
48680CPR      COMMONS USED --
48690CPR
48700CPR          AFIL
48710CPR          SIZE
48720CPR          WORK
48730CPR
48740CPR      FUNCTIONS/ROUTINES CALLED --
48750CPR
48760CPR          EINCOR
48770CPR          RDSPF
48780CPR          FNDELM
48790CPR
48800CPR      FILES USED --
48810CPR
48820CPR          INOT6      S
48830CPR
48840CPR      LIBRARIES ACCESSED --
48850CPR      NONE
48860CPR
48870CPR
48880CPR      LOCAL VARIABLES --
48890CPR
48900CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
48910CPR
48920CPR
48930CPR      SPECIAL REMARKS/INSTRUCTIONS --
48940CPR          HOST = H6000
48950CPR
48960CPR
48970CPR=====
• 49580      END
```

TITLE= DATE =07/15/85 TIME = 23 42 PAGE=47

```
49600C=====
49620      SUBROUTINE RDSPF(NN,X,Y,Z,IERC)
49640C=====
49660CPR
49670CPR=====
49680CPR
49690CPR
49700CPR      ROUTINE--          RDSPF
49710CPR
49720CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFEO
49730CPR      DATE/REVISED BY--
49740CPR
49750CPR      FUNCTION/PURPOSE--
49760CPR
49770CPR      THIS ROUTINE READS THE STRESS POINT COORD INFO
49780CPR      FROM THE STRESS POINT NODE FILE
49790CPR
49800CPR
49810CPR
49820CPR      CALLING ARGUMENTS --
49830CPR          NAME          ATTRIBUTES      DEFINITION
49840CPR
49850CPR          NN          (0)          NODE NAME
49860CPR          X          (0)          X COORD
49870CPR          Y          (0)          Y COORD
49880CPR          Z          (0)          Z COORD
49890CPR          IERC        (0)          ERROR CODES
49900CPR                                     0 -- NO ERRORS
49910CPR                                     1 -- END OF FILE
49920CPR                                     2 -- READ ERROR
49930CPR
49940CPR      COMMONS USED --
49950CPR
49960CPR          AFIL
49970CPR
49980CPR      FUNCTIONS/ROUTINES CALLED --
49990CPR
50000CPR
50010CPR      FILES USED --
50020CPR
50030CPR          ISPNF      S
50040CPR
50050CPR      LIBRARIES ACCESSED --
50060CPR      NONE
50070CPR
50080CPR
50090CPR      LOCAL VARIABLES --
50100CPR
50110CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
50120CPR
50130CPR
50140CPR      SPECIAL REMARKS/INSTRUCTIONS --
50150CPR          HOST = H6000
50160CPR
50170CPR
50180CPR=====
50440      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=48

```
50460C=====
50480      SUBROUTINE FILREW(IFC)
50500C=====
50520CPR
50530CPR=====
50540CPR
50550CPR
50560CPR      ROUTINE--          FILREW
50570CPR
50580CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFEO
50590CPR      DATE/REVISED BY--
50600CPR
50610CPR      FUNCTION/PURPOSE--
50620CPR
50630CPR      THIS ROUTINE REWINDS FILES
50640CPR
50650CPR
50660CPR      CALLING ARGUMENTS --
50670CPR          NAME          ATTRIBUTES      DEFINITION
50680CPR
50690CPR          IFC          (I)          FILE CODE
50700CPR
50710CPR      COMMONS USED --
50720CPR
50730CPR
50740CPR      FUNCTIONS/ROUTINES CALLED --
50750CPR
50760CPR      FILES USED --
50770CPR
50780CPR
50790CPR      LIBRARIES ACCESSED --
50800CPR      NONE
50810CPR
50820CPR
50830CPR      LOCAL VARIABLES --
50840CPR
50850CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
50860CPR
50870CPR
50880CPR      SPECIAL REMARKS/INSTRUCTIONS --
50890CPR          HOST = H6000
50900CPR
50910CPR
50920CPR=====
50960      END
```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=49

```

50980C=====
51000      SUBROUTINE EINCOR(ENX,NELMN,NELMX)
51020C=====
51040CPR
51050CPR=====
51060CPR
51070CPR
51080CPR      ROUTINE--          EINCOR
51090CPR
51100CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFEO
51110CPR      DATE/REVISED BY--
51120CPR
51130CPR      FUNCTION/PURPOSE--
51140CPR
51160CPR      COORDS INTO CORE FROM THE RANDOM ELM FILE
51170CPR      THE DATA STORED IN CGRE IS --
51180CPR          ELM NUMBER
51190CPR          XMIN
51200CPR          YMIN
51210CPR          ZMIN
51220CPR          XMAX
51230CPR          YMAX
51240CPR          ZMAX
51250CPR
51260CPR
51270CPR      CALLING ARGUMENTS --
51280CPR          NAME          ATTRIBUTES      DEFINITION
51290CPR
51300CPR          ENX          (O)          ARRAY CONTAINING WINDOW INFO
51310CPR          NELMN        (I)          MIN ELM NAME
51320CPR          NELMX        (I)          MAX ELM NAME
51330CPR
51340CPR      COMMONS USED --
51350CPR
51360CPR          AFIL
51370CPR          ELMDAT
51380CPR
51390CPR      FUNCTIONS/ROUTINES CALLED --
51400CPR
51410CPR          RWELMS
51420CPR
51430CPR      FILES USED --
51440CPR
51450CPR
51460CPR      LIBRARIES ACCESSED --
51470CPR      NONE
51480CPR
51490CPR
51500CPR      LOCAL VARIABLES --
51510CPR
51520CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
51530CPR
51540CPR
51550CPR      SPECIAL REMARKS/INSTRUCTIONS --
51560CPR          HOST = H6000
51570CPR
51580CPR
51590CPR=====
52000      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=50

```

52020C=====
52040      SUBROUTINE FNDELM(NSP,XP,YP,ZP,IFOND)
52060C=====
52080CPR
52090CPR=====
52100CPR
52110CPR
52120CPR      ROUTINE--          FNDELM
52130CPR
52140CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFED
52150CPR      DATE/REVISED BY--
52160CPR
52170CPR      FUNCTION/PURPOSE--
52180CPR
52190CPR      THIS ROUTINE DOES THE ACTUAL SEARCH THRU THE HEAT
52200CPR      TRANSFER ELMS TO FIND WHICH ELM CONTAINS THE STRESS
52210CPR      POINT
52220CPR      IT CALLS THE REQUIRED ROUTINES TO GET THE ELM
52230CPR      COORDINATES,CHECKS TO SEE IF THE STRESS POINT
52240CPR      IS WITHIN THE ELM AND IF IT IS COMPUTES
52250CPR      AND WRITES THE WEIGHTING COEFFS FOR THIS STRESS
52260CPR      POINT WITH RESPECT TO THE VERTICES OF THE
52270CPR      HEAT TRANSFER ELEMENT
52280CPR
52290CPR
52300CPR
52310CPR      CALLING ARGUMENTS --
52320CPR          NAME          ATTRIBUTES      DEFINITION
52330CPR
52340CPR          NSP            (I)              STRESS NODE NAME
52350CPR          XP             (I) >>
52360CPR          YP            (I) >>>>        STRESS POINT COORDS
52370CPR          ZP             (I) >>
52380CPR          IFOND           (O)              INDICATOR
52390CPR                                          0 -- DID NOT FIND CONTAINING ELM
52400CPR                                          1 -- DID      FIND CONTAINING ELM
52410CPR
52420CPR      COMMONS USED --
52430CPR
52440CPR          AFIL
52450CPR          ELMDAT
52460CPR          SIZE
52470CPR          SPDAT
52480CPR
52490CPR      FUNCTIONS/ROUTINES CALLED --
52500CPR
52510CPR          CKWIND
52520CPR          GETWND
52530CPR          I3DSF
52540CPR          RWSPCF
52550CPR
52560CPR      FILES USED --
52570CPR
52580CPR
52590CPR      LIBRARIES ACCESSED --
52600CPR      NONE
52610CPR
52620CPR
52630CPR      LOCAL VARIABLES --
52640CPR
52650CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
52660CPR
52670CPR
52680CPR      SPECIAL REMARKS/INSTRUCTIONS --
52690CPR          HOST = H6000
52700CPR
52710CPR

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=51

52720CPR=-----
54420 END

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=52

```

54440C=====
54460      SUBROUTINE GETWND(IDONE,IWHR)
54480C=====
54500CPR
54510CPR=====
54520CPR
54530CPR
54540CPR      ROUTINE--          GETWND
54550CPR
54560CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFED
54570CPR      DATE/REVISED BY--
54580CPR
54590CPR      FUNCTION/PURPOSE--
54600CPR
54620CPR      OF THE HEAT TRANSFER ELMS AND THE COORDS OF THE
54630CPR      CORNERS OF THE ELM
54640CPR      IT CAN GET THIS INFO FROM FILE OR FROM CORE
54650CPR
54660CPR
54670CPR
54680CPR      CALLING ARGUMENTS --
54690CPR      NAME          ATTRIBUTES      DEFINITION
54700CPR
54710CPR      IDONE      (I)      SEQUENTIAL POINTER TO HT ELM DATA
54720CPR      IWHR      (I)      INDICATOR
54730CPR                                1 -- GET ONLY WINDOW COORDS
54740CPR                                2 -- GET CORNER COORDS
54750CPR
54760CPR      COMMONS USED --
54770CPR
54780CPR      AFIL
54790CPR      ELMDAT
54800CPR      SIZE
54810CPR      WORK
54820CPR
54830CPR      FUNCTIONS/ROUTINES CALLED --
54840CPR
54850CPR      RWELMS
54860CPR
54870CPR      FILES USED --
54880CPR
54890CPR
54900CPR      LIBRARIES ACCESSED --
54910CPR      NONE
54920CPR
54930CPR
54940CPR      LOCAL VARIABLES --
54950CPR
54960CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
54970CPR
54980CPR      THE COORD DATA IS PASSED BACK TO THE ROUTINE -FNDELM-
54990CPR      VIA THE COMMON CALLED ELMDAT IN THE ARRAY AED
55000CPR
55010CPR
55020CPR      SPECIAL REMARKS/INSTRUCTIONS --
55030CPR      HOST = H6000
55040CPR
55050CPR
55060CPR=====
55730      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=53

```

55750C=====
55770      SUBROUTINE CKWIND(XP,YP,ZP,V,INOT)
55790C=====
55810CPR
55820CPR=====
55830CPR
55840CPR
55850CPR      ROUTINE--          CKWIND
55860CPR
55870CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFED
55880CPR      DATE/REVISED BY--
55890CPR
55900CPR      FUNCTION/PURPOSE--
55910CPR
55920CPR      THIS ROUTINE CHECKS TO SEE IF THE COORDS OF THE
55930CPR      STRESS POINT ARE WITHIN THE WINDOW OF THE HEAT TRANSFER
55940CPR      ELEMENT
55950CPR
55960CPR
55970CPR
55980CPR      CALLING ARGUMENTS --
55990CPR      NAME          ATTRIBUTES      DEFINITION
56000CPR
56010CPR      XP >>
56020CPR      YP >>>>      (I)          COORDS OF STRESS POINT
56030CPR      ZP >>
56050CPR      COORDS OF HEAT TRAN ELM
56060CPR      INOT      (O)          IN - OUT INDICATOR
56070CPR      0 -- OUT OF WINDOW
56080CPR      1 -- IN WINDOW
56090CPR
56100CPR      COMMONS USED --
56110CPR
56120CPR
56130CPR      FUNCTIONS/ROUTINES CALLED --
56140CPR
56150CPR
56160CPR      FILES USED --
56170CPR
56180CPR
56190CPR      LIBRARIES ACCESSED --
56200CPR      NONE
56210CPR
56220CPR
56230CPR      LOCAL VARIABLES --
56240CPR
56250CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
56260CPR
56270CPR
56280CPR      SPECIAL REMARKS/INSTRUCTIONS --
56290CPR      HOST = H6000
56300CPR
56310CPR
56320CPR=====
56610      END

```

```

56620      SUBROUTINE MKRGF
56640C=====
56650CPR
56660CPR
56670CPR      ROUTINE--      MKRGF
56680CPR
56690CPR      DATE/WRITTEN BY--  8/04/82      RJ MAFFEO
56700CPR      DATE/REVISED BY-- 4/04/83      RJ MAFFEO
56710CPR              REASON --      ADDED MIN MAX WINDOW LOGIC
56720CPR      DATE/REVISED BY--  8/30/83      SR MARTIN
56730CPR              REASON -- CHANGED I/O TO READ NEW INPUT FILE FORM
56740CPR
56750CPR      FUNCTION/PURPOSE--
56760CPR      THIS ROUTINE DOES THE PROCESSING OF THE NEUTRAL
56770CPR      NODE AND ELEMENT FILES INTO THE INTERNAL DATA BASE FILES
56780CPR      FOR THIS VERSION OF 3D TITS.
56790CPR      IT CALLS RWNODS TO PROCESS THE NODAL INFO.
56800CPR      IT CALLS RWELMS TO PROCESS THE ELM INFO
56810CPR      IT CALLS PROFAC TO PROCESS THE FACE INFORMATION
56820CPR
56830CPR
56840CPR      CALLING SEQUENCE--  CALL MKRGF
56850CPR
56860CPR      CALLING ARGUMENTS--  NONE
56870CPR              NAME      ATTRIBUTES      DEFINITION
56880CPR
56890CPR
56900CPR      FILES USED--
56910CPR      IFL3  -- RANDOM FILE TO STORE THE NODAL COORDINATES
56920CPR      IFL4  -- RANDOM FILE TO STORE THE ELM ATTRIBUTES
56930CPR      INTITS -- HEAT TRANSFER OUTPUT FILE
56940CPR      INOT6 -- OUTPUT FILE FOR MESSAGES
56950CPR      IERF  -- ERROR REMARK FILE
56960CPR
56970CPR
56980CPR      COMMONS USED--
56990CPR      AFIL   -- FILECODE COMMON
57000CPR      ELMDAT -- ELM ATTRIBUTE COMMON
57010CPR      SIZE   -- SIZE VARIABLES COMMON
57020CPR      CNTLFL -- INFO WHICH DESCRIBES ALL ELEMENTS
57030CPR
57040CPR
57050CPR
57060CPR      FUNCTIONS/ROUTINES CALLED--
57070CPR      RWNODS
57080CPR      HEADEL
57090CPR      RWELMS
57100CPR      READPS
57110CPR      PROFAC
57120CPR      EMINMX
57130CPR
57140CPR
57150CPR      LIBRARIES ACCESSED--
57160CPR      NONE
57170CPR
57180CPR
57190CPR      LOCAL VARIABLES--
57200CPR      NNODMX -- MAXIMUM NODE NAME
57210CPR      NNODMN -- MINIMUM NODE NAME
57220CPR      NUMNOD -- NUMBER OF NODES
57230CPR      NELMX  -- MAXIMUM ELM NAME
57240CPR      NELMN  -- MINIMUM ELM NAME
57250CPR      NUME   -- NUMBER OF ELEMENTS
57260CPR      CRD(45) -- ARRAY OF COORDINATES FOR THE 15 POINTS IN A ELM
57270CPR      IFACPC -- PACKED THTD FACE NUMBERS
57280CPR      IE(8)  -- ARRAY OF CONNECTIVITY FOR ELM CORENER POINTS
57290CPR      AED(56) -- ELM ATTRIBUTE ARRAY

```

TITLE= DATE =07/15/85 TIME = 23 42 PAGE=55

```
57300CPR          THIS ARRAY CONTAINS
57310CPR          NE,CRD,IE,IFACPC,VOL
57320CPR  NE      -- ELM NAME
57330CPR  VOL      -- VOLUME OF THE ELM
57340CPR  XYZ(3)   -- COORDINATE ARRAY FOR EACH NODE
57350CPR
57360CPR
57370CPR  SPECIAL COMMENTS ABOUT THIS ROUTINE
57380CPR  NOTE THE 15 POINTS FOR EACH ELM ARE
57390CPR  8 CORNERS , 6 MIDSIDES AND 1 CENTROID
57400CPR
57410CPR
57420CPR  SPECIAL REMARKS/INSTRUCTIONS--
57430CPR  HOST = H6000
57440CPR
57450CPR          *****
58690  END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=56

```
58700      SUBROUTINE PROFAC
58720C=====
58730CPR
58740CPR
58750CPR      ROUTINE--    PROFAC
58760CPR
58770CPR      DATE/WRITTEN BY--  8/04/82    RJ MAFFEO
58780CPR      DATE/REVISED BY--
58790CPR
58800CPR      FUNCTION/PURPOSE--
58810CPR      THIS ROUTINE READS THE PACKED THTD FACE NUMBER FROM THE
58820CPR      END OF THE NEUTRAL ELEMENT FILE AND STORES THEM ON
58830CPR      THE INTERNAL DATA BASE FILE
58840CPR      THIS DATA BASE FILE IS RANDOM FILE IFL4,
58850CPR      THE ELEMENTAL ATTRIBUTE FILE
58860CPR      NOTE THE PACKED FACE NUMBERS EXIST ON THE HEAT
58870CPR      TRANSFER OUTPUT FILE
58880CPR      IN RECORDS OF THE FORM- <ELMT #> <FACE INFO>
58890CPR
58900CPR
58910CPR      CALLING SEQUENCE--    CALL PROFAC
58920CPR
58930CPR      CALLING ARGUMENTS--  NONE
58940CPR          NAME      ATTRIBUTES      DEFINITION
58950CPR
58960CPR
58970CPR      FILES USED--
58980CPR          IFL4 -- RANDOM ELEMENT ATTRIBUTE FILE
58990CPR          INTITS -- HEAT TRANSFER OUTPUT FILE (INPUT TO TITAN)
59000CPR          IERF -- ERROR REMARK FILE
59010CPR
59020CPR
59030CPR      COMMONS USED--
59040CPR          AFIL  -- FILECODE COMMON
59050CPR          ELMDAT -- ELEMENT ATTRIBUTE COMMON
59060CPR          SIZE  -- SIZE OF PROBLEM VARIABLES IN THIS COMMON
59070CPR
59080CPR
59090CPR
59100CPR      FUNCTIONS/ROUTINES CALLED--
59110CPR          RWELMS
59120CPR
59130CPR
59140CPR      LIBRARIES ACCESSED--
59150CPR          NONE
59160CPR
59170CPR      LOCAL VARIABLES--
59180CPR          NRED    --- NUMMBER OF NUMBERS READ FROM NEF FOR ONE SET OF
59190CPR                      ELEMENT FACE DEFINITIONS (LIMIT IS 200)
59200CPR          IFACPC --- PACKED FACE NUMBER
59210CPR          NCNT    --- TOTAL NUMBER OF ELMS THAT HAVE FACES DEFINED
59220CPR
59230CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
59240CPR
59250CPR
59260CPR      SPECIAL REMARKS/INSTRUCTIONS--
59270CPR          HOST = H6000
59280CPR
59290CPR          *****
59890      END
```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=57

```

59910C=====
59930      SUBROUTINE SKIP (IFC,ISETS)
59950C=====
59960CPR
59980CPR=====
59990CPR
60000CPR
60010CPR      ROUTINE--          SKIP
60020CPR
60030CPR      DATE/WRITTEN BY-- 08/31/83          SR MARTIN
60040CPR      DATE/REVISED BY--
60050CPR
60060CPR      FUNCTION/PURPOSE--
60070CPR          THIS ROUTINE ALLOWS THE CALLING ROUTINE TO SKIP PAST
60080CPR          UNNEEDED TIMESTEP AND TEMPERATURE DATA
60090CPR
60100CPR      CALLING ARGUMENTS --
60110CPR          NAME          ATTRIBUTES          DEFINITION
60120CPR
60130CPR          IFC          INPUT,INTEGER          FILE CODE
60140CPR          ISETS          INPUT,INTEGER          NUMBER OF SETS TO SKIP OVER
60150CPR
60160CPR      COMMONS USED --
60170CPR          CNTLFL
60180CPR          SIZE
60190CPR
60200CPR
60210CPR      FUNCTIONS/ROUTINES CALLED --
60220CPR
60230CPR
60240CPR      FILES USED --
60250CPR
60260CPR          FC          S
60270CPR
60280CPR      LIBRARIES ACCESSED --
60290CPR      NONE
60300CPR
60310CPR
60320CPR      LOCAL VARIABLES --
60330CPR
60340CPR          DUMMY (CHARACTER) USED TO DUMMY READ LINES IN THE FILE
60350CPR
60360CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
60370CPR
60380CPR
60390CPR      SPECIAL REMARKS/INSTRUCTIONS --
60400CPR          HOST = H6000
60410CPR
60420CPR
60430CPR=====
60630      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=58

```
60650C=====
60670      SUBROUTINE HTFIO (IFC,DATA,ICNT,ISTAT)
60690C=====
60710CPR=====
60720CPR
60730CPR
60740CPR      ROUTINE--          HTFIO
60750CPR
60760CPR      DATE/WRITTEN BY-- 09/13/83      SR MARTIN
60770CPR      DATE/REVISED BY--
60780CPR
60790CPR      FUNCTION/PURPOSE--
60800CPR      THIS ROUTINE READS INFO FROM THE HEAT TRANSFER INPUT FILE
60810CPR
60820CPR      CALLING ARGUMENTS --
60830CPR          IFC (INPUT/INTEGER) FILE CODE OF FILE TO READ FROM
60840CPR          DATA (OUTPUT/REAL) ARRAY OF DATA PASSED BACK
60850CPR          ICNT (INPUT/INTEGER) AMOUNT OF DATA IN 'DATA' ARRAY
60860CPR          ISTAT (OUTPUT/INTEGER) RETURN STATUS, VALUES ARE--
60870CPR              O=OK, 1=END OF FILE, 2=READ ERROR
60880CPR
60890CPR      COMMONS USED --
60900CPR
60910CPR
60920CPR      FUNCTIONS/ROUTINES CALLED --
60930CPR
60940CPR
60950CPR      FILES USED --
60960CPR
60970CPR          IFC      S
60980CPR
60990CPR      LIBRARIES ACCESSED --
61000CPR      NONE
61010CPR
61020CPR
61030CPR      LOCAL VARIABLES --
61040CPR
61050CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
61060CPR
61070CPR
61080CPR      SPECIAL REMARKS/INSTRUCTIONS --
61090CPR          HOST = H6000
61100CPR
61110CPR
61120CPR=====
61270      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=59

```
61280      SUBROUTINE RDCNTL
61290CPR
61300CPR=====
61310CPR
61320CPR
61330CPR      ROUTINE--          RDCNTL
61340CPR
61350CPR      DATE/WRITTEN BY-- 09/13/83      SR MARTIN
61360CPR      DATE/REVISED BY--
61370CPR
61380CPR      FUNCTION/PURPOSE--
61390CPR          READ USER HEADER INFORMATION AND FILE COUNTERS
61400CPR
61410CPR
61420CPR
61430CPR
61440CPR      CALLING ARGUMENTS --
61450CPR          NAME          ATTRIBUTES      DEFINITION
61460CPR
61470CPR
61480CPR      COMMONS USED --
61490CPR
61500CPR          AFIL
61510CPR          CNTLFL
61520CPR          CNTLTM
61530CPR          SIZE
61540CPR
61550CPR      FUNCTIONS/ROUTINES CALLED --
61560CPR
61570CPR          HTFIO
61580CPR
61590CPR      FILES USED --
61600CPR
61610CPR          INTITS      S
61620CPR          INOT6      S
61630CPR
61640CPR      LIBRARIES ACCESSED --
61650CPR          NONE
61660CPR
61670CPR
61680CPR      LOCAL VARIABLES --
61690CPR
61700CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
61710CPR
61720CPR
61730CPR      SPECIAL REMARKS/INSTRUCTIONS --
61740CPR          HOST = H6000
61750CPR
61760CPR
61770CPR=====
62640      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=60

```
62650      SUBROUTINE TFLSET
62660CPR
62670CPR=====
62680CPR
62690CPR
62700CPR      ROUTINE--          TFLSET
62710CPR
62720CPR      DATE/WRITTEN BY-- 09/14/83      SR MARTIN
62730CPR      DATE/REVISED BY--
62740CPR
62750CPR      FUNCTION/PURPOSE--
62760CPR          THIS ROUTINE REWINDS THE HEAT TRANSFER FILE AND REPOSITIONS
62770CPR          THE FILE POINTER TO THE FIRST RECORD OF THE TIME/TEMP
62780CPR          INFORMATION
62790CPR
62800CPR
62810CPR
62820CPR
62830CPR      CALLING ARGUMENTS --
62840CPR          NAME          ATTRIBUTES      DEFINITION
62850CPR
62860CPR
62870CPR      COMMONS USED --
62880CPR
62890CPR          AFIL
62900CPR          CNTLTM
62910CPR
62920CPR      FUNCTIONS/ROUTINES CALLED --
62930CPR
62940CPR
62950CPR      FILES USED --
62960CPR
62970CPR          INTITS      S
62980CPR
62990CPR      LIBRARIES ACCESSED --
63000CPR      NONE
63010CPR
63020CPR
63030CPR      LOCAL VARIABLES --
63040CPR
63050CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
63060CPR
63070CPR
63080CPR      SPECIAL REMARKS/INSTRUCTIONS --
63090CPR          HOST = H6000
63100CPR
63110CPR
63120CPR=====
63420      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=61

```
6344OC=====
6346O      SUBROUTINE GENTMP
6348OC=====
6350OCPR
6351OCPR=====
6352OCPR
6353OCPR
6354OCPR      ROUTINE--          GENTMP
6355OCPR
6356OCPR      DATE/WRITTEN BY-- 09/26/83      RJ MAFFEO
6357OCPR      DATE/REVISED BY--
6358OCPR
6359OCPR      FUNCTION/PURPOSE--
6360OCPR
6361OCPR      THIS ROUTINE IS THE MINI-EXEC USED TO CALL ALL
6362OCPR      ROUTINES ASSOCIATED WITH COMPUTING THE FINAL VALUES
6363OCPR      OF TEMPERATURE FOR EACH STRESS NODE
6364OCPR      IT CAN DEAL WITH BOTH FINITE DIFFERENCE AND
6365OCPR      FINITE ELEMNT HEAT TRANSFER CODES
6366OCPR      IF THE HEAT TRANSFER CODE IS FINITE DIFFERENCE
6367OCPR      THE TEMPERATURES OF THE CORNERS OF THE HEAT TRANSFER
6368OCPR      ELEMENTS ARE FOUND VIA CALLS FROM THIS ROUTINE
6369OCPR
6370OCPR      CALLING ARGUMENTS --      NONE
6371OCPR
6372OCPR
6373OCPR      COMMONS USED --
6374OCPR
6375OCPR          MAFCOM
6376OCPR          AFIL
6377OCPR          MAFCOM
6378OCPR          SIZE
6379OCPR          MAFCOM
6380OCPR          WORK
6381OCPR          MAFCOM
6382OCPR          CNTLTM
6383OCPR          CNTLFL
6384OCPR
6385OCPR      FUNCTIONS/ROUTINES CALLED --
6386OCPR
6387OCPR          GETMPE
6388OCPR          SETFLP
6389OCPR          RDTEMP
6390OCPR          CWCNRT
6391OCPR          ETCORE
6392OCPR          NTCORE
6393OCPR          STSTMP
6394OCPR
6395OCPR      FILES USED --
6396OCPR
6397OCPR
6398OCPR      LIBRARIES ACCESSED --
6399OCPR      NONE
6400OCPR
6401OCPR
6402OCPR      LOCAL VARIABLES --
6403OCPR
6404OCPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
6405OCPR
6406OCPR
6407OCPR      SPECIAL REMARKS/INSTRUCTIONS --
6408OCPR          HOST = H6000
6409OCPR
6410OCPR
6411OCPR=====
6528O      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=62

```

65300C=====
65320      SUBROUTINE STSTMP(TEMP,TIMVAL)
65340C=====
65360CPR
65370CPR=====
65380CPR
65390CPR
65400CPR      ROUTINE--          STSTMP
65410CPR
65420CPR      DATE/WRITTEN BY-- 09/28/83      RJ MAFFEO
65430CPR      DATE/REVISED BY--
65440CPR
65450CPR      FUNCTION/PURPOSE--
65460CPR
65470CPR      THIS ROUTINE READS THE COEFF INFO FOR EACH STRESS POINT
65480CPR      (VIA CALL RWSPCF) GETS THE CORNER TEMPS FROM THE TEMP ARRAY
65490CPR      AND COMPUTES THE TEMPERATURE OF THE STRESS POINT BY
65500CPR      SUMMING THE COEFFS TIMES THE CORNER TEMPS
65510CPR
65520CPR      CALLING ARGUMENTS --
65530CPR          NAME          ATTRIBUTES      DEFINITION
65540CPR
65550CPR          TEMP          (I)          ARRAY CONTAINING TEMPS OF HEAT
65560CPR                      TRANSFER CORNERS
65570CPR          TIMVAL      (I)          TIME STEP VALUE
65580CPR
65590CPR      COMMONS USED --
65600CPR
65610CPR          MAFCOM
65620CPR          AFIL
65630CPR          MAFCOM
65640CPR          SIZE
65650CPR          SPDAT
65660CPR
65670CPR      FUNCTIONS/ROUTINES CALLED --
65680CPR
65690CPR          RWSPCF
65700CPR          WRSTMP
65710CPR
65720CPR      FILES USED --
65730CPR
65740CPR
65750CPR      LIBRARIES ACCESSED --
65760CPR      NONE
65770CPR
65780CPR
65790CPR      LOCAL VARIABLES --
65800CPR
65810CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
65820CPR
65830CPR
65840CPR      SPECIAL REMARKS/INSTRUCTIONS --
65850CPR          HOST = H6000
65860CPR
65870CPR
65880CPR=====
66460      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=63

```

66480C=====
66500      SUBROUTINE WRSTMP(NAME,TEMP,TIMVAL,NDONE)
66520C=====
66540CPR
66550CPR=====
66560CPR
66570CPR
66580CPR      ROUTINE--          WRSTMP
66590CPR
66600CPR      DATE/WRITTEN BY-- 09/28/83      RJ MAFFEO
66610CPR      DATE/REVISED BY--
66620CPR
66630CPR      FUNCTION/PURPOSE--
66640CPR
66650CPR      THIS ROUTINE WRITES THE STRESS POINT TEMPERATURES
66660CPR      TO THE OUTPUT FILE (IN NEUTRAL TEMPERATURE FILE FORM)
66670CPR
66680CPR      CALLING ARGUMENTS --
66690CPR          NAME          ATTRIBUTES          DEFINITION
66700CPR          ----          -
66710CPR          NAME          (I)          STRESS POINT LABEL
66720CPR          TEMP          (I)          TEMPERATURE OF STRESS POINT
66730CPR          TIMVAL        (I)          TIME VALUE OF STEP
66740CPR          NDONE         (I)          # OF NODES WRITTEN
66750CPR
66760CPR      COMMONS USED --
66770CPR
66780CPR          MAFCOM
66790CPR          AFIL
66800CPR
66810CPR      FUNCTIONS/ROUTINES CALLED -- NONE
66820CPR
66830CPR
66840CPR      FILES USED --
66850CPR
66860CPR          IOUTF      S
66870CPR
66880CPR      LIBRARIES ACCESSED --
66890CPR      NONE
66900CPR
66910CPR
66920CPR      LOCAL VARIABLES --
66930CPR
66940CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
66950CPR
66960CPR
66970CPR      SPECIAL REMARKS/INSTRUCTIONS --
66980CPR          HOST = H6000
66990CPR
67000CPR
67010CPR=====
67280      END

```

TITLE=

DATE =07/15/85

TIME = 23.42

PAGE=64

```
67300C=====
67320      SUBROUTINE ETCORE(ETMP)
67340C=====
67360CPR
67370CPR=====
67380CPR
67390CPR
67400CPR      ROUTINE--          ETCORE
67410CPR
67420CPR      DATE/WRITTEN BY-- 09/29/83      RJ MAFFEO
67430CPR      DATE/REVISED BY--
67440CPR
67450CPR      FUNCTION/PURPOSE--
67460CPR
67470CPR      THIS ROUTINE TRANSFERS THE FINITE DIFFERENCE ELEMENTAL
67480CPR      CORNER TEMPERATURES FROM SCRATCH FILE TO CORE
67490CPR
67500CPR      CALLING ARGUMENTS --
67510CPR          NAME          ATTRIBUTES      DEFINITION
67520CPR
67530CPR          ETMP          (O)              ARRAY CONTAINING ELM CORNER TEMPS
67540CPR
67550CPR      COMMONS USED --
67560CPR
67570CPR          AFIL
67580CPR          SIZE
67590CPR
67600CPR      FUNCTIONS/ROUTINES CALLED --
67610CPR
67620CPR          FILREW
67630CPR
67640CPR      FILES USED --
67650CPR
67660CPR          ICRNTF      S
67670CPR          IERF       S
67680CPR
67690CPR      LIBRARIES ACCESSED --
67700CPR      NONE
67710CPR
67720CPR
67730CPR      LOCAL VARIABLES --
67740CPR
67750CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
67760CPR
67770CPR
67780CPR      SPECIAL REMARKS/INSTRUCTIONS --
67790CPR          HOST = H6000
67800CPR
67810CPR
67820CPR=====
68230      END
```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=65

```

68250C=====
68270      SUBROUTINE NTCORE(TEMP)
68290C=====
68310CPR
68320CPR=====
68330CPR
68340CPR
68350CPR      ROUTINE--          MAIN
68360CPR
68370CPR      DATE/WRITTEN BY-- 09/19/83      SR MARTIN
68380CPR      DATE/REVISED BY--
68390CPR
68400CPR      FUNCTION/PURPOSE--
68410CPR          THIS ROUTINE ASSUMES THE FILE POINTER IS POINTING AT THE
68420CPR          FIRST NODE NUMBER, TEMPERATURE PAIR OF A
68430CPR          TIMESTEP IT READS IN EACH PAIR AND STORES THE
68440CPR          TEMPERATURE (FROM DATA(2)) INTO THE 'TEMP' ARRAY
68450CPR          IN POSITION <NODE NUMBER> (IE--DATA(1))
68460CPR          THE RESULT BEING THAT, FOR SAY NODE 13, THE TEMPERATURE
68470CPR          WOULD BE IN RECORD 13 OF THE 'TEMP' ARRAY
68480CPR          UPON RETURN, THE FILE POINTER WILL POINT
68490CPR          TO THE NEXT TIMESTEP VALUE
68500CPR
68510CPR
68520CPR
68530CPR
68540CPR      CALLING ARGUMENTS --
68550CPR          NAME      ATTRIBUTES      DEFINITION
68560CPR          TEMP      REAL,OUTPUT      ARRAY WHICH WILL CONTAIN
68570CPR                                     TEMPERATURE VALUES NOTE--
68580CPR                                     TEMP(I) WILL CONTAIN THE
68590CPR                                     TEMPERATURE VALUE FOR NODE 'I'
68600CPR
68610CPR
68620CPR      COMMONS USED --
68630CPR
68640CPR          AFIL
68650CPR          SIZE
68660CPR
68670CPR      FUNCTIONS/ROUTINES CALLED --
68680CPR
68690CPR          HTFIO
68700CPR
68710CPR      FILES USED --
68720CPR
68730CPR
68740CPR      LIBRARIES ACCESSED --
68750CPR      NONE
68760CPR
68770CPR
68780CPR      LOCAL VARIABLES --
68790CPR          DATA (REAL,ARRAY) USED TO RETRIEVE THE NODE NUMBER,
68800CPR          TEMPERATURE PAIR FORM THE INPUT FILE
68810CPR          NODNUM (INTEGER) USED TO STORE THE NODE NUMBER
68820CPR          AS AS INTEGER AFTER RETRIEVING IT VIA THE
68830CPR          'DATA' ARRAY.
68840CPR
68850CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
68860CPR
68870CPR
68880CPR      SPECIAL REMARKS/INSTRUCTIONS --
68890CPR          HOST = H6000
68900CPR
68910CPR
68920CPR=====
69200      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=66

```
69220C=====
69240      SUBROUTINE WRCRNT(TBAR,NODE)
69260C=====
69280CPR
69290CPR=====
69300CPR
69310CPR
69320CPR      ROUTINE--          WRCRNT
69330CPR
69340CPR      DATE/WRITTEN BY-- 09/29/83      RJ MAFFEO
69350CPR      DATE/REVISED BY--
69360CPR
69370CPR      FUNCTION/PURPOSE--
69380CPR
69390CPR      THIS ROUTINE WRITES THE FINITE DIFFERENCE CORNER
69400CPR      TEMPS TO THE SCRATCH FILE.
69410CPR
69420CPR      CALLING ARGUMENTS --
69430CPR          NAME          ATTRIBUTES      DEFINITION
69440CPR
69450CPR          TBAR          (I)          VALUE OF TEMPERATURE
69460CPR          NODE          (I)          H T ELM CORNER NAME
69470CPR
69480CPR      COMMONS USED --
69490CPR
69500CPR          AFIL
69510CPR
69520CPR      FUNCTIONS/ROUTINES CALLED --
69530CPR
69540CPR
69550CPR      FILES USED --
69560CPR
69570CPR          ICRNTF  S
69580CPR
69590CPR      LIBRARIES ACCESSED --
69600CPR      NONE
69610CPR
69620CPR
69630CPR      LOCAL VARIABLES --
69640CPR
69650CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
69660CPR
69670CPR
69680CPR      SPECIAL REMARKS/INSTRUCTIONS --
69690CPR          HOST = H6000
69700CPR
69710CPR
69720CPR=====
69930      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=67

```

69950C=====
69970      SUBROUTINE HTFPAS(IFC,NFP,IFILP)
69990C=====
70010CPR
70020CPR=====
70030CPR
70040CPR
70050CPR      ROUTINE--          HTFPAS
70060CPR
70070CPR      DATE/WRITTEN BY-- 09/29/83      RJ MAFFEO
70080CPR      DATE/REVISED BY--
70090CPR
70100CPR      FUNCTION/PURPOSE--
70110CPR
70120CPR      THIS ROUTINE SKIPS PAST PORTIONS OF THE
70130CPR      HEAT TRANSFER INPUT FILE
70140CPR      IT CURRENTLY ONLY CAN SKIP PAST THE TIME-TEMPERATURE
70150CPR      PORTION(IFILP=6)
70160CPR
70170CPR      CALLING ARGUMENTS --
70180CPR          NAME          ATTRIBUTES      DEFINITION
70190CPR
70200CPR          IFC          (I)      FILE CODE OF FILE TO BE SKIPED
70210CPR          NFP          (I)      # OF PARTITIONS TO SKIP
70220CPR          IFILP        (I)      PORTION OF FILE(DATA TYPE)TO BE SKIPPED
70230CPR                                     IFILP=1 THRU 5 NOT CURRENTLY USED
70240CPR                                     IFILP=6 SKIP TIME-TEMPERATURE RECORDS
70250CPR
70260CPR      COMMONS USED --
70270CPR
70280CPR          SIZE
70290CPR
70300CPR      FUNCTIONS/ROUTINES CALLED --
70310CPR
70320CPR
70330CPR      FILES USED --
70340CPR
70350CPR          IFC          S
70360CPR
70370CPR      LIBRARIES ACCESSED --
70380CPR      NONE
70390CPR
70400CPR
70410CPR      LOCAL VARIABLES --
70420CPR
70430CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
70440CPR
70450CPR
70460CPR      SPECIAL REMARKS/INSTRUCTIONS --
70470CPR          HOST = H6000
70480CPR
70490CPR
70500CPR=====
70840      END

```

TITLE= DATE =07/15/85 TIME = 23 42 PAGE=68

```
70860C=====
70880      SUBROUTINE SORT(SEEDS,LAX,NDUM,IKEY)
70900C=====
70910CPR
70920CPR=====
70930CPR
70940CPR
70950CPR      ROUTINE--          SORT
70960CPR
70970CPR      DATE/WRITTEN BY--  ????????      GE SORT ROUTINE
70980CPR      DATE/REVISED BY--
70990CPR
71000CPR      FUNCTION/PURPOSE--
71010CPR
71020CPR      THIS ROUTINE SORTS 1 DIMENSIONAL ARRAYS
71030CPR
71040CPR      CALLING ARGUMENTS --
71050CPR          NAME          ATTRIBUTES      DEFINITION
71060CPR
71070CPR          SEEDS      (I/O)      ARRAY TO BE SORTED
71080CPR          LAX        (I)        NUMBER OF ITEMS IN ARRAY
71090CPR          NDUM      DUMMY      DUMMY
71100CPR          IKEY      DUMMY      DUMMY
71110CPR
71120CPR      COMMONS USED --      NONE
71130CPR
71140CPR      FUNCTIONS/ROUTINES CALLED -- NONE
71150CPR
71160CPR      FILES USED --      NONE
71170CPR
71180CPR      LIBRARIES ACCESSED --
71190CPR      NONE
71200CPR
71210CPR      LOCAL VARIABLES --
71220CPR
71230CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
71240CPR
71250CPR
71260CPR      SPECIAL REMARKS/INSTRUCTIONS --
71270CPR          HOST = H6000
71280CPR
71290CPR
71300CPR=====
71550      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=69

```

71570C=====
71590      SUBROUTINE HTICON(IHTIC)
71610C=====
71630CPR
71640CPR
71650CPR
71660CPR      ROUTINE--          HTICON
71670CPR
71680CPR      DATE/WRITTEN BY-- 01/14/84      RJ MAFFED
71690CPR      DATE/REVISED BY--
71700CPR
71710CPR      FUNCTION/PURPOSE--
71720CPR      THIS ROUTINE CONVERTS FROM THE HEAT TRANSFER OUTPUT FORM
71730CPR      TO THE INTERNAL 3D TITAN HEAT TRANSFER INPUT FORM
71740CPR      VALID HEAT TRANSFER INPUT CODES ARE
71750CPR
71760CPR      0   --- INTERNAL 3D TITAN HEAT TRANSFER INPUT FORM
71770CPR      1   --- SINDA HEAT TRANSFER INPUT FORM
71780CPR      2   --- MARC HEAT TRANSFER INPUT FORM (FORMATTED)
71790CPR      3   --- MARC HEAT TRANSFER INPUT FORM (SEQUENTIAL BINARY)
71800CPR      4   --- THTD HEAT TRANSFER INPUT FORM
71810CPR
71820CPR
71830CPR
71840CPR      CALLING ARGUMENTS --
71850CPR          NAME      ATTRIBUTES      DEFINITION
71860CPR
71870CPR          IHTIC      (I)          HEAT TRANSFER INPUT CODE
71880CPR
71890CPR      COMMONS USED --
71900CPR
71910CPR          AFIL
71920CPR
71930CPR      FUNCTIONS/ROUTINES CALLED --
71940CPR
71950CPR          SINTIT
71960CPR          MARCRD
71970CPR
71980CPR      FILES USED --
71990CPR
72000CPR          IERF      S
72010CPR
72020CPR      LIBRARIES ACCESSED --
72030CPR      NONE
72040CPR
72050CPR
72060CPR      LOCAL VARIABLES --
72070CPR
72080CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
72090CPR
72100CPR
72110CPR      SPECIAL REMARKS/INSTRUCTIONS --
72120CPR          HOST = H6000
72130CPR
72140CPR
72150CPR=====
72700      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=70

```

72710      SUBROUTINE MARCRD(IFTYP,NSTEPS)
72720CPR
72730CPR=====
72740CPR
72750CPR
72760CPR      ROUTINE--          MARCRD
72770CPR
72780CPR      DATE/WRITTEN BY-- 01/11/84      RD MCCLAIN
72790CPR      DATE/REVISED BY--
72800CPR
72810CPR      FUNCTION/PURPOSE--
72820CPR
72830CPR          READ MARC POST TAPE AND WRITE TRANSFER MODULE
72840CPR          INPUT FILE
72850CPR
72860CPR      CALLING ARGUMENTS --
72870CPR          NAME          ATTRIBUTES      DEFINITION
72880CPR
72890CPR          IFTYP          INPUT          FILE TYPE OF MARC POST TAPE
72900CPR                                     O=FORMATTED, 1=SEQ  BINARY
72910CPR          NSTEPS        INPUT          NUMBER OF STEPS TO READ FROM
72920CPR                                     MARC POST TAPE
72930CPR
72940CPR      COMMONS USED --
72950CPR          AFIL -- FILE CODES
72960CPR
72970CPR      FUNCTIONS/ROUTINES CALLED --
72980CPR
72990CPR          READR          READS DATA FROM POST TAPE
73000CPR          SKPREC       SKIPS RECORDS ON POST TAPE
73010CPR
73020CPR      FILES USED --
73030CPR
73040CPR          INTITS      S   TRANSFER MODULE INPUT FILE
73050CPR          IHTIN      S   MARC INPUT FILE (SEQ BIN OR FORMATTED)
73060CPR
73070CPR      LIBRARIES ACCESSED --
73080CPR          NONE
73090CPR
73100CPR
73110CPR      LOCAL VARIABLES --
73120CPR
73130CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
73140CPR
73150CPR
73160CPR      SPECIAL REMARKS/INSTRUCTIONS --
73170CPR          HOST = H6000
73180CPR
73190CPR
73200CPR=====
74770      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=71

```

74810      SUBROUTINE READR(LFN,DATA,NDATA,CHAR,NWRDS,ITYPE,IFTYP,IEND)
74840CPR
74850CPR=====
74860CPR
74870CPR
74880CPR      ROUTINE--          READR
74890CPR
74900CPR      DATE/WRITTEN BY-- 01/11/84      RD MCCLAIN
74910CPR      DATE/REVISED BY--
74920CPR
74930CPR      FUNCTION/PURPOSE--
74940CPR
74950CPR          READ FLOATING POINT,INTEGER, OR CHARACTER DATA
74960CPR          FROM MARC POST TAPE
74970CPR
74980CPR      CALLING ARGUMENTS --
74990CPR          NAME          ATTRIBUTES      DEFINITION
75000CPR
75010CPR          LFN          INPUT          FILE CODE TO READ FROM
75020CPR          DATA        OUTPUT         FLOATING POINT ARRAY
75030CPR          NDATA        OUTPUT         INTEGER ARRAY
75040CPR          CHAR         OUTPUT         CHARACTER ARRAY
75050CPR          NWRDS        INPUT          NUMBER OF ITEMS TO READ
75060CPR          ITYPE        INPUT          TYPE OF DATA TO READ
75070CPR                                     1=LOAD FP DATA INTO ARRAY DATA
75080CPR                                     2=LOAD INTEGER DATA INTO ARRAY NDATA
75090CPR                                     3=LOAD CHAR DATA INTO ARRAY CHAR
75100CPR          IFTYP        INPUT          FILE TYPE
75110CPR                                     O=FORMATTED
75120CPR                                     1=SEQUENTIAL BINARY
75130CPR
75140CPR      COMMONS USED --
75150CPR          AFIL -- FILE CODES
75160CPR
75170CPR      FUNCTIONS/ROUTINES CALLED --
75180CPR
75190CPR
75200CPR      FILES USED --
75210CPR
75220CPR          LFN      S      FILE CODE TO BE READ
75230CPR          INOT6   S      STANDARD OUTPUT FILE
75240CPR
75250CPR      LIBRARIES ACCESSED --
75260CPR      NONE
75270CPR
75280CPR
75290CPR      LOCAL VARIABLES --
75300CPR
75310CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
75320CPR
75330CPR
75340CPR      SPECIAL REMARKS/INSTRUCTIONS --
75350CPR          HOST = H6000
75360CPR
75370CPR
75380CPR=====
75870      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=72

```

75900      SUBROUTINE SKPREC(IFILE,NUMREC,IFTYP)
75930CPR
75940CPR=====
75950CPR
75960CPR
75970CPR      ROUTINE--          SKPREC
75980CPR
75990CPR      DATE/WRITTEN BY-- 01/11/84      RD MCCLAIN
76000CPR      DATE/REVISED BY--
76010CPR
76020CPR      FUNCTION/PURPOSE--
76030CPR
76040CPR          SKIP RECORDS ON A SEQUENTIAL FILE
76050CPR
76060CPR
76070CPR      CALLING ARGUMENTS --
76080CPR          NAME          ATTRIBUTES      DEFINITION
76090CPR
76100CPR          IFILE          INPUT          FILE CODE TO SKIP RECORDS ON
76110CPR          NUMREC          INPUT          NUMBER OF RECORDS TO SKIP
76120CPR          IFTYP          INPUT          FILE TYPE OF MARC POST TAPE
76130CPR                                         O=FORMATTED, 1=BINARY
76140CPR
76150CPR      COMMONS USED --  NONE
76160CPR
76170CPR
76180CPR      FUNCTIONS/ROUTINES CALLED --
76190CPR
76200CPR
76210CPR      FILES USED --
76220CPR
76230CPR
76240CPR      LIBRARIES ACCESSED --
76250CPR      NONE
76260CPR
76270CPR
76280CPR      LOCAL VARIABLES --
76290CPR
76300CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
76310CPR
76320CPR
76330CPR      SPECIAL REMARKS/INSTRUCTIONS --
76340CPR          HOST = H6000
76350CPR
76360CPR
76370CPR=====
76690      END

```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=73

```

76710C=====
76730      SUBROUTINE SINTIT
76750C=====
76770CPR
76780CPR=====
76790CPR
76800CPR
76810CPR      ROUTINE--          SINTIT
76820CPR
76830CPR      DATE/WRITTEN BY-- 01/09/84      RN PITTMAN
76840CPR      DATE/REVISED BY--
76850CPR
76860CPR      FUNCTION/PURPOSE--
76870CPR          MAIN EXECUTIVE ROUTINE FOR CREATION OF SINDA-TO-3DTITAN
76880CPR          INTERFACE FILE
76890CPR
76900CPR      CALLING ARGUMENTS --
76910CPR          NAME          ATTRIBUTES      DEFINITION
76920CPR          -----NONE-----
76930CPR
76940CPR      COMMONS USED --
76950CPR          WORK - STORES INTERNAL ARRAY DATA
76960CPR          -----
76970CPR                  TEMP          ARRAY OF ELEMENTAL TEMPERATURES
76980CPR                                CORRESPONDING TO 'ITELM'
76990CPR                  IPKFAC        ARRAY OF PACKED FACE NUMBERS
77000CPR                                CORRESPONDING TO 'IGELM'
77010CPR                  IGELM         ARRAY OF ELEMENTS WITH GEOMETRY
77020CPR                  ITEL          ARRAY OF ELEMENTS WITH TEMPERATURES
77030CPR
77040CPR      AFIL - FILE CODES
77050CPR      -----
77060CPR          ELEMENTS          DEFINITION
77070CPR          -----
77080CPR          IHTIN              SINDA TEMPERATURE OUTPUT FILE (INPUT)
77090CPR          ISINGF           SINDA GEOMETRY FILE (INPUT)
77100CPR          INTITS          INTERFACE FILE (OUTPUT)
77110CPR          IERF             HARDCOPY FILE (OUTPUT)
77120CPR
77130CPR      KNTDAT - COUNTER DATA
77140CPR      -----
77150CPR          ELEMENTS          DEFINITION
77160CPR          -----
77170CPR          NUMNOD              NUMBER OF NODES (CORNER POINTS)
77180CPR                                IN MODEL
77190CPR          NUMEWG             NUMBER OF ELEMENTS WITH GEOMETRY
77200CPR          NUMTIM            NUMBER OF SOLUTION TIMES
77210CPR          NNPE              NUMBER OF NODES PER ELEMENT
77220CPR          ECODE              ELEMENT CODE
77230CPR                                =1 - 2D MODEL (NNPE=4)
77240CPR                                =11 - 3D MODEL (NNPE=8)
77250CPR          ACODE             ANALYSIS CODE
77260CPR                                =1 - FINITE DIFFERENCE ANALYSIS
77270CPR          NUMEWT             NUMBER OF ELEMENTS WITH TEMPERATURES
77280CPR          NUMSKP            NUMBER OF RECORDS TO SKIP ON 'IHTIN'
77290CPR                                PRIOR TO BEGINNING OF TEMPERATURE DATA
77300CPR          NUMFUL            NUMBER OF FULL RECORDS OF DATA
77320CPR          NUMREM            NUMBER OF WORDS ON FINAL RECORD
77340CPR
77350CPR      FUNCTIONS/ROUTINES CALLED --
77360CPR          DATINT
77370CPR          PRCSOF
77380CPR          PRCGEO
77390CPR          WRTNTF
77400CPR
77410CPR      FILES USED --
77420CPR          NONE

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=74

7743OCPR
7744OCPR LIBRARIES ACCESSED --
7745OCPR NONE
7746OCPR
7747OCPR LOCAL VARIABLES --
7748OCPR NONE
7749OCPR
7750OCPR SPECIAL COMMENTS ABOUT THIS ROUTINE --
7751OCPR PROGRAM IS CURRENTLY DIMENSIONED TO PERMIT A MAXIMUM OF
7752OCPR 5000 ELEMENTS WITH GEOMETRY AND 5000 ELEMENTS WITH
7753OCPR TEMPERATURES
7754OCPR
7755OCPR SPECIAL REMARKS/INSTRUCTIONS --
7756OCPR HOST = HIS
7757OCPR
7758OCPR
7759OCPR-----
78060 END

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=75

```

78100      SUBROUTINE DATINT
78140CPR
78150CPR=====
78160CPR
78170CPR
78180CPR      ROUTINE--          DATINT
78190CPR
78200CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
78210CPR      DATE/REVISED BY--
78220CPR
78230CPR      FUNCTION/PURPOSE--
78240CPR          THIS ROUTINE INITIALIZES ALL OF THE GLOBAL COMMON VARIABLES
78250CPR
78260CPR      CALLING ARGUMENTS --
78270CPR          NAME          ATTRIBUTES      DEFINITION
78280CPR          -----NONE-----
78290CPR
78300CPR      COMMONS USED --
78310CPR          WORK -- STORES INTERNAL ARRAY DATA
78320CPR          AFIL -- FILE CODES (MAIN)
78330CPR          KNTDAT -- COUNTER DATA (MAIN)
78340CPR
78350CPR      FUNCTIONS/ROUTINES CALLED --
78360CPR          NONE
78370CPR
78380CPR      FILES USED --
78390CPR          NONE
78400CPR
78410CPR      LIBRARIES ACCESSED --
78420CPR          NONE
78430CPR
78440CPR      LOCAL VARIABLES --
78450CPR          NONE
78460CPR
78470CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
78480CPR          NONE
78490CPR
78500CPR      SPECIAL REMARKS/INSTRUCTIONS --
78510CPR          HOST = H6000
78520CPR
78530CPR
78540CPR=====
79020C      .  END OF SUBROUTINE 'DATINT'
79050      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=76

```
79090      SUBROUTINE PRC Sof
79130CPR
79140CPR=====
79150CPR
79160CPR
79170CPR      ROUTINE--          PRC Sof
79180CPR
79190CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
79200CPR      DATE/REVISED BY--
79210CPR
79220CPR      FUNCTION/PURPOSE--
79230CPR          THIS ROUTINE CONTROLS THE INITIAL PROCESSING OF THE
79240CPR          'SINDA OUTPUT FILE' (THE FILE WHICH CONTAINS THE
79250CPR          RESULTS OF THE SINDA ANALYSIS)
79260CPR
79270CPR      CALLING ARGUMENTS --
79280CPR          NAME          ATTRIBUTES      DEFINITION
79290CPR          -----NONE-----
79300CPR
79310CPR      COMMONS USED --
79320CPR          AFIL -- FILE CODES (MAIN)
79330CPR
79340CPR      FUNCTIONS/ROUTINES CALLED --
79350CPR          GETEWT
79360CPR          GTNUMT
79370CPR
79380CPR      FILES USED --
79390CPR          NONE
79400CPR
79410CPR      LIBRARIES ACCESSED --
79420CPR          NONE
79430CPR
79440CPR      LOCAL VARIABLES --
79450CPR          NONE
79460CPR
79470CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
79480CPR          NONE
79490CPR
79500CPR      SPECIAL REMARKS/INSTRUCTIONS --
79510CPR          HOST = H6000
79520CPR
79530CPR
79540CPR=====
79770C      . END OF SUBROUTINE 'PRC Sof'
79800      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=77

```

79840      SUBROUTINE GETEWT
79880CPR
79890CPR=====
79900CPR
79910CPR
79920CPR      ROUTINE--          GETEWT
79930CPR
79940CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
79950CPR      DATE/REVISED BY--
79960CPR
79970CPR      FUNCTION/PURPOSE--
79980CPR          1  READS THE SINDA ELEMENT NUMBERS FROM 'IHTIN' INTO 'ITEML'
79990CPR          2  READS FROM 'IHTIN' THE TOTAL NUMBER OF SINDA ELEMENTS
80000CPR              WITH TEMPERATURES INTO 'NUMEWT'
80010CPR          3  COUNTS THE NUMBER OF RECORDS PRECEEDING THE FIRST TEMP-
80020CPR              ERATURE SOLUTION ON 'IHTIN' AND STORES THIS VALUE IN
80030CPR              'NUMSKP'.
80040CPR
80050CPR      CALLING ARGUMENTS --
80060CPR          NAME          ATTRIBUTES      DEFINITION
80070CPR          -----NONE-----
80080CPR
80090CPR      COMMONS USED --
80100CPR          WORK  --STORES INTERNAL ARRAY DATA
80110CPR          AFIL  -- FILE CODES (MAIN)
80120CPR          KNTDAT -- COUNTER DATA (MAIN)
80130CPR
80140CPR      FUNCTIONS/ROUTINES CALLED --
80150CPR          ERRPRT
80160CPR
80170CPR      FILES USED --
80180CPR          IHTIN   (S)   SINDA OUTPUT FILE (INPUT TO THIS ROUTINE)
80190CPR
80200CPR      LIBRARIES ACCESSED --
80210CPR          NONE
80220CPR
80230CPR      LOCAL VARIABLES --
80240CPR          IPOS  -  POSITION OF THE BEGINNING OF EACH GROUP OF 10
80250CPR                  ELEMENT NUMBERS IN THE 'ITEML' ARRAY
80260CPR          ***NOTE***
80270CPR          IN THIS ROUTINE, THE GLOBAL VARIABLES 'NUMFUL' AND NUMREM'
80280CPR          ARE REDEFINED AS FOLLOWS-
80290CPR              NUMFUL - NUMBER OF 'FULL' 10-WORD RECORDS OF SINDA
80300CPR                  ELEMENT NUMBERS CONTAINED ON 'IHTIN'
80310CPR              NUMREM - NUMBER OF ELEMENT NUMBERS REMAINING ON FINAL
80320CPR                  RECORD OF ELEMENT NUMBERS
80330CPR
80340CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
80350CPR          NONE
80360CPR
80370CPR      SPECIAL REMARKS/INSTRUCTIONS --
80380CPR          HOST = H6000
80390CPR
80400CPR
80410CPR=====
81210CPR      END OF SUBROUTINE 'GETEWT'
81240      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=78

```

81280      SUBROUTINE GTNUMT
81320CPR
81330CPR-----
81340CPR
81350CPR
81360CPR      ROUTINE--          GTNUMT
81370CPR
81380CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
81390CPR      DATE/REVISED BY--
81400CPR
81410CPR      FUNCTION/PURPOSE--
81420CPR          THIS ROUTINE COUNTS THE TOTAL NUMBER OF SOLUTION BLOCKS
81430CPR          CONTAINED IN THE 'SINDA OUTPUT FILE' AND STORES THIS VALUE
81440CPR          IN THE GLOBAL VARIABLE 'NUMTIM'
81450CPR
81460CPR      CALLING ARGUMENTS --
81470CPR          NAME          ATTRIBUTES      DEFINITION
81480CPR          -----NONE-----
81490CPR
81500CPR      COMMONS USED --
81510CPR          WORK  --STORES  INTERNAL ARRAY DATA
81520CPR          AFIL  -- FILE CODES (MAIN)
81530CPR          KNTDAT -- COUNTER DATA (MAIN)
81540CPR
81550CPR      FUNCTIONS/ROUTINES CALLED --
81560CPR          ERRPRT
81570CPR
81580CPR      FILES USED --
81590CPR          IHTIN  (S)      SINDA OUTPUT FILE (INPUT TO THIS PROGRAM)
81600CPR
81610CPR      LIBRARIES ACCESSED --
81620CPR          NONE
81630CPR
81640CPR      LOCAL VARIABLES --
81650CPR          DUMMY - VARIABLE USED IN DUMMY READS TO PASS THROUGH EACH
81660CPR                  SOLUTION BLOCK TO THE NEXT
81670CPR          KNT   - CURRENT COUNT OF THE NUMBER OF SOLUTION BLOCKS
81680CPR                  ENCOUNTERED ON 'IHTIN'
81690CPR          ***NOTE***
81700CPR          IN THIS ROUTINE, THE GLOBAL VARIABLES 'NUMFUL' AND 'NUMREM'
81710CPR          ARE REDEFINED AS FOLLOWS-
81720CPR          NUMFUL - NUMBER OF 'FULL' 6-WORD RECORDS OF TEMPERATURE
81730CPR                  DATA IN EACH SOLUTION BLOCK ON 'IHTIN'
81740CPR          NUMREM - NUMBER OF TEMPERATURES ON FINAL RECORD OF EACH
81750CPR                  SOLUTION BLOCK
81760CPR          THESE DEFINITIONS ARE NOT ALTERED FURTHER IN THIS PROGRAM
81770CPR
81780CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
81790CPR          EACH TEMPERATURE SOLUTION BLOCK ON 'IHTIN' IS PASSED THROUGH
81800CPR          WITH DUMMY READS UNTIL THE END OF FILE IS ENCOUNTERED
81810CPR          THE SOLUTION COUNTER, 'KNT', IS INCREMENTED BY ONE AT THE
81820CPR          BEGINNING OF EACH BLOCK
81830CPR
81840CPR      SPECIAL REMARKS/INSTRUCTIONS --
81850CPR          HOST = H6000
81860CPR
81870CPR
81880CPR-----
82550CPR      . . END OF SUBROUTINE 'GTNUMT'
82580      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=79

```
82620      SUBROUTINE PRCGED
82660CPR
82670CPR=====
82680CPR
82690CPR
82700CPR      ROUTINE--          PRCGED
82710CPR
82720CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
82730CPR      DATE/REVISED BY--
82740CPR
82750CPR      FUNCTION/PURPOSE--
82760CPR          1  READS THE VALUES CONTAINED IN THE 'DIRECTORY' RECORD OF
82770CPR              THE 'SINDA GEOMETRY FILE' INTO THE GLOBAL COMMON VALUES
82780CPR                  'NUMNOD', 'NUMEWG', AND 'NNPE'
82790CPR          2  SETS THE VALUE FOR 'ECODE' BASED UPON THE VALUE OF 'NNPE'
82800CPR
82810CPR      CALLING ARGUMENTS --
82820CPR          NAME          ATTRIBUTES      DEFINITION
82830CPR          -----NONE-----
82840CPR
82850CPR      COMMONS USED --
82860CPR          AFIL -- FILE CODES (MAIN)
82870CPR          KNTDAT -- CCOUNTER DATA (MAIN)
82880CPR
82890CPR      FUNCTIONS/ROUTINES CALLED --
82900CPR          ERRPRT
82910CPR
82920CPR      FILES USED --
82930CPR          ISINGF      (S)      SINDA GEOMETRY FILE (INPUT)
82940CPR
82950CPR      LIBRARIES ACCESSED --
82960CPR          NONE
82970CPR
82980CPR      LOCAL VARIABLES --
82990CPR          TITLE - CHARACTER ARRAY INTO WHICH THE 'ISINGF' TITLE RECORD
83000CPR                  IS READ
83010CPR
83020CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
83030CPR          NONE
83040CPR
83050CPR      SPECIAL REMARKS/INSTRUCTIONS --
83060CPR          HOST = H6000
83070CPR
83080CPR
83090CPR=====
83560C ... END OF SUBROUTINE 'PRCGED'
83590      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=80

```
83630      SUBROUTINE WRTNTF
83670CPR
83680CPR=====
83690CPR
83700CPR
83710CPR      ROUTINE--          WRTNTF
83720CPR
83730CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
83740CPR      DATE/REVISED BY--
83750CPR
83760CPR      FUNCTION/PURPOSE--
83770CPR          THIS ROUTINE CONTROLS THE WRITING OF THE 'INTERFACE FILE'.
83780CPR
83790CPR      CALLING ARGUMENTS --
83800CPR          NAME          ATTRIBUTES      DEFINITION
83810CPR          -----NONE-----
83820CPR
83830CPR      COMMONS USED --
83840CPR          WORKS  --STORES  INTERNAL ARRAY DATA
83850CPR          AFIL  -- FILE CODES (MAIN)
83860CPR          KNTDAT -- COUNTER DATA (MAIN)
83870CPR
83880CPR      FUNCTIONS/ROUTINES CALLED --
83890CPR          WRTDIR
83900CPR          WRTGEO
83910CPR          WRTEMP
83920CPR          ERRPRT
83930CPR
83940CPR      FILES USED --
83950CPR          INTITS   (S)   INTERFACE FILE (OUTPUT)
83960CPR          ISINGF   (S)   SINDA GEOMETRY FILE (INPUT)
83970CPR          IHTIN   (S)   SINDA OUTPUT FILE (INPUT)
83980CPR
83990CPR      LIBRARIES ACCESSED --
84000CPR          NONE
84010CPR
84020CPR      LOCAL VARIABLES --
84030CPR          DUMMY - VARIABLE USED IN 'DUMMY' READS TO SKIP OVER ALL
84040CPR                  DATA ON 'IHTIN' PRECEEDING THE TEMPERATURE
84050CPR                  SOLUTION BLOCKS
84060CPR
84070CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
84080CPR          NONE
84090CPR
84100CPR      SPECIAL REMARKS/INSTRUCTIONS --
84110CPR          HOST = H6000
84120CPR
84130CPR
84140CPR=====
84850CPR      .  END OF SUBROUTINE 'WRTNTF'
84880      END
```


TITLE= DATE =07/15/85 TIME = 23 42 PAGE=81

```

84920      SUBROUTINE WRTDIR
84960CPR
84970CPR=====
84980CPR
84990CPR
85000CPR      ROUTINE--          WRTDIR
85010CPR
85020CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
85030CPR      DATE/REVISED BY--
85040CPR
85050CPR      FUNCTION/PURPOSE--
85060CPR          THIS ROUTINE WRITES THE TITLE (RECORDS 1-4) AND DIRECTORY
85070CPR          (RECORD 5) RECORDS OF THE 'INTERFACE FILE'
85080CPR
85090CPR      CALLING ARGUMENTS --
85100CPR          NAME          ATTRIBUTES      DEFINITION
85110CPR          -----NONE-----
85120CPR
85130CPR      COMMONS USED --
85140CPR          WORK      --STORES INTERNAL ARRAY DATA
85150CPR          AFIL -- FILE CODES (MAIN)
85160CPR          KNTDAT -- COUNTER DATA (MAIN)
85170CPR
85180CPR      FUNCTIONS/ROUTINES CALLED --
85190CPR          ERRPRT
85200CPR
85210CPR      FILES USED --
85220CPR          INTITS      (S)      INTERFACE FILE (OUTPUT)
85230CPR          ISINGF      (S)      SINDA GEOMETRY FILE (INPUT)
85240CPR
85250CPR      LIBRARIES ACCESSED --
85260CPR          NONE
85270CPR
85280CPR      LOCAL VARIABLES --
85290CPR          TITLE - 80-CHARACTER TITLE CONTAINED ON RECORD 1 OF 'ISINGF'
85300CPR                  AND WRITTEN ON RECORD 4 OF 'INTITS'
85310CPR          USRTTL - 80-CHARACTER ARRAY USED TO 'DUMMY-OUT' RECORDS
85320CPR                  1-3 OF 'INTITS'
85330CPR
85340CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
85350CPR          NONE
85360CPR
85370CPR      SPECIAL REMARKS/INSTRUCTIONS --
85380CPR          HOST = H6000
85390CPR
85400CPR
85410CPR=====
86090C      . END OF SUBROUTINE 'WRTDIR'
86120      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=82

```

86160      SUBROUTINE WRTGEO
86200CPR
86210CPR=====
86220CPR
86230CPR
86240CPR      ROUTINE--          WRTGEO
86250CPR
86260CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
86270CPR      DATE/REVISED BY--
86280CPR
86290CPR      FUNCTION/PURPOSE--
86300CPR          THIS ROUTINE WRITES THE GEOMETRICAL DATA (COORDINATE,
86310CPR          CONNECTIVITY, AND FACES) ON THE INTERFACE FILE
86320CPR
86330CPR      CALLING ARGUMENTS --
86340CPR          NAME          ATTRIBUTES      DEFINITION
86350CPR          -----NONE-----
86360CPR
86370CPR      COMMONS USED --
86380CPR          WORK  --STORES INTERNAL ARRAY DATA
86390CPR          AFIL -- FILE CODES (MAIN)
86400CPR          KNTDAT -- COUNTER DATA (MAIN)
86410CPR
86420CPR      FUNCTIONS/ROUTINES CALLED --
86430CPR          ERRPR
86440CPR
86450CPR      FILES USED --
86460CPR          ISINGF  (S)    SINDA GEOMETRY FILE (INPUT)
86470CPR          INTITS  (S)    INTERFACE FILE (OUTPUT)
86480CPR
86490CPR      LIBRARIES ACCESSED --
86500CPR          NONE
86510CPR
86520CPR      LOCAL VARIABLES --
86530CPR          DUMMY - USED IN 'DUMMY READ' TO POSITION 'ISINGF' AT
86540CPR                  BEGINNING OF COORDINATE DATA
86550CPR          N      - ARRAY OF NODE NUMBERS DEFINING CONNECTIVITY
86560CPR                  OF 'IGELM(I)'
86570CPR          NODE   - CURRENT NODE (CORNER POINT) NUMBER
86580CPR          X      - X-COORDINATE OF 'NODE'
86590CPR          Y      - Y-COORDINATE OF 'NODE'
86600CPR          Z      - Z-COORDINATE OF 'NODE'
86610CPR
86620CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
86630CPR          NONE
86640CPR
86650CPR      SPECIAL REMARKS/INSTRUCTIONS --
86660CPR          HOST = H6000
86670CPR
86680CPR
86690CPR=====
87490C ..  END OF SUBROUTINE 'WRTGEO'
87520      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=83

```

87560      SUBROUTINE WRTEMP
87600CPR
87610CPR=====
87620CPR
87630CPR
87640CPR      ROUTINE--          WRTEMP
87650CPR
87660CPR      DATE/WRITTEN BY-- 01/11/84      RN PITTMAN
87670CPR      DATE/REVISED BY--
87680CPR
87690CPR      FUNCTION/PURPOSE--
87700CPR          THIS ROUTINE WRITES THE TIME AND TEMPERATURE DATA ON THE
87710CPR          'INTERFACE FILE'
87720CPR
87730CPR      CALLING ARGUMENTS --
87740CPR          NAME          ATTRIBUTES      DEFINITION
87750CPR          -----NONE-----
87760CPR
87770CPR      COMMONS USED --
87780CPR          WORK      --STORES INTERNAL ARRAY DATA
87790CPR          AFIL -- FILE CODES (MAIN)
87800CPR          KNTDAT -- COUNTER DATA (MAIN)
87810CPR
87820CPR      FUNCTIONS/ROUTINES CALLED --
87830CPR          ERRPR
87840CPR
87850CPR      FILES USED --
87860CPR          IHTIN   (S)   SINDA OUTPUT FILE (INPUT)
87870CPR          INTITS  (S)   INTERFACE FILE (OUTPUT)
87880CPR
87890CPR      LIBRARIES ACCESSED --
87900CPR          NONE
87910CPR
87920CPR      LOCAL VARIABLES --
87930CPR          KNT  - POSITION IN THE 'TEMP' ARRAY OF THE NEXT GROUP OF
87940CPR                SIX TEMPERATURES TO BE WRITTEN TO 'INTITS'
87950CPR          TIME - CURRENT VALUE OF TIME READ FROM 'IHTIN' FOR THIS
87960CPR                TEMPERATURE BLOCK
87970CPR
87980CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
87990CPR          NONE
88000CPR
88010CPR      SPECIAL REMARKS/INSTRUCTIONS --
88020CPR          HOST = H6000
88030CPR
88040CPR
88050CPR=====
88840C      END OF SUBROUTINE 'WRTEMP'
88870      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=84

```

88910      SUBROUTINE ERRPRT (IROUT)
88950CPR
88960CPR=====
88970CPR
88980CPR
88990CPR      ROUTINE--          ERRPRT
89000CPR
89010CPR      DATE/WRITTEN BY-- 01/12/84      RN PITTMAN
89020CPR      DATE/REVISED BY--
89030CPR
89040CPR      FUNCTION/PURPOSE--
89050CPR          THIS ROUTINE PROCESSES ALL FATAL ERROR CONDITIONS BY
89060CPR          PRINTING AN EXPLANATION OF THE ERROR ON THE HARDCOPY
89070CPR          OUTPUT FILE AND TERMINATING PROGRAM EXECUTION
89080CPR
89090CPR      CALLING ARGUMENTS --
89100CPR          NAME          ATTRIBUTES          DEFINITION
89110CPR          ----          -
89120CPR          IROUT      INTEGER/INPUT      ROUTING VARIABLE FOR APPROPRIATE
89130CPR                                     ERROR MESSAGE
89140CPR
89150CPR      COMMONS USED --
89160CPR          AFIL -- FILE CODES (MAIN)
89170CPR
89180CPR      FUNCTIONS/ROUTINES CALLED --
89190CPR          NONE
89200CPR
89210CPR      FILES USED --
89220CPR          IERF      (S)      HARDCOPY FILE (OUTPUT)
89230CPR
89240CPR      LIBRARIES ACCESSED --
89250CPR          NONE
89260CPR
89270CPR      LOCAL VARIABLES --
89280CPR          NONE
89290CPR
89300CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE --
89310CPR          NONE
89320CPR
89330CPR      SPECIAL REMARKS/INSTRUCTIONS --
89340CPR          HOST = H6000
89350CPR
89360CPR
89370CPR=====
91950      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=85

```

91970C=====
91990      SUBROUTINE TEMFOR(ISPOC)
92010C=====
92030CPR
92040CPR=====
92050CPR
92060CPR
92070CPR      ROUTINE--          TEMFOR
92080CPR
92090CPR      DATE/WRITTEN BY-- 01/14/84      RJ MAFFED
92100CPR      DATE/REVISED BY--
92110CPR
92120CPR      FUNCTION/PURPOSE--
92130CPR      THIS ROUTINE CALLS ALL THE REQUIRED ROUTINES TO FORMAT THE
92140CPR      OUTPUT TEMPERATURES INTO FORMED NEEDED FOR THE STRESS ANALYSIS
92150CPR      PROGRAM
92160CPR          VALID CODES ARE
92170CPR              0 --- NEUTRAL TEMPERATURE OUTPUT FILE ONLY
92180CPR              1 --- NASTRAN FORMATTED TEMPERATURES
92190CPR              2,3 --- MARC      FORMATTED TEMPERATURES
92200CPR
92210CPR      CALLING ARGUMENTS --
92220CPR          NAME      ATTRIBUTES      DEFINITION
92230CPR
92240CPR          ISPOC      (I)          STRESS PROGRAM OUTPUT CODE
92250CPR
92260CPR      COMMONS USED --
92270CPR
92280CPR          AFIL
92290CPR          SIZE
92300CPR
92310CPR      FUNCTIONS/ROUTINES CALLED --
92320CPR
92330CPR          TPNAST
92340CPR          MARCTO
92350CPR
92360CPR      FILES USED --
92370CPR
92380CPR          IERF      S
92390CPR
92400CPR      LIBRARIES ACCESSED --
92410CPR      NONE
92420CPR
92430CPR
92440CPR      LOCAL VARIABLES --
92450CPR
92460CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
92470CPR
92480CPR
92490CPR      SPECIAL REMARKS/INSTRUCTIONS --
92500CPR          HOST = H6000
92510CPR
92520CPR
92530CPR=====
93110      END

```

TITLE= DATE =07/15/85 TIME = 23 42 PAGE=86

```
93140      SUBROUTINE TPNAST(NUMNOD,NUMTMP)
93170CPR
93180CPR-----
93190CPR
93200CPR
93210CPR      ROUTINE--          TPNAST
93220CPR
93230CPR      DATE/WRITTEN BY-- 01/14/84      RD MCCLAIN
93240CPR      DATE/REVISED BY--
93250CPR
93260CPR      FUNCTION/PURPOSE--
93270CPR
93280CPR          WRITE NASTRAN BULK DATA TEMPERATURE CARDS
93290CPR
93300CPR
93310CPR      CALLING ARGUMENTS --
93320CPR          NAME          ATTRIBUTES      DEFINITION
93330CPR
93340CPR          NUMNOD          INPUT          NUMBER OF NODES ON NEUT TEMP FILE
93350CPR          NUMTMP          INPUT          NUMBER OF TEMPERATURE SETS
93360CPR
93370CPR      COMMONS USED --
93380CPR
93390CPR          AFIL -- FILE CODES
93400CPR
93410CPR      FUNCTIONS/ROUTINES CALLED --
93420CPR
93430CPR
93440CPR      FILES USED --
93450CPR
93460CPR          IOUTF      S      NEUTRAL OUTPUT FILE
93470CPR          ITEMP      S      NASTRAN BULK DATA FILE
93480CPR
93490CPR      LIBRARIES ACCESSED --
93500CPR      NONE
93510CPR
93520CPR
93530CPR      LOCAL VARIABLES --
93540CPR
93550CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
93560CPR
93570CPR
93580CPR      SPECIAL REMARKS/INSTRUCTIONS --
93590CPR          HOST = H6000
93600CPR
93610CPR
93620CPR-----
93910      END
```

```

93930C=====
93950      SUBROUTINE I3DSF(CMNMX,CRD,XP,YP,ZP,NSP,
93970CPR
93980CPR=====
93990CPR
94000CPR
94010CPR      ROUTINE--          I3DSF
94020CPR
94030CPR      DATE/WRITTEN BY-- 11/28/83      R J  MAFFEO
94040CPR      DATE/REVISED BY--
94050CPR
94060CPR      FUNCTION/PURPOSE--
94070CPR      THIS ROUTINE COMPUTES THE VALUES OF THE LOCAL COORDINATES
94080CPR      OF A STRESS POINT WRT THE VERTICES OF THE 8-NODED HEAT
94090CPR      TRANSFER ELEMENT.IF THE STRESS POINT LIES IN THE ELM
94100CPR      IT COMPUTES THE 8 WEIGHTING COEFFICIENTS FOR THIS POINT
94110CPR      IF THE POINT LIES OUTSIDE THE ELM IT COMPUTES THE SURFACE
94120CPR      COORDINATES OF THE POINT PROJECTED IN THE LOCAL COORD SYSTEM
94130CPR      AND COMPUTES THE WEIGHTING COEFFS BASED ON THESE SURFACE
94140CPR      COORDINATES IT ALSO COMPUTES THE DISTANCE THE POINT IS AWAY
94150CPR      FROM THE SURFACE
94160CPR
94170CPR      CALLING ARGUMENTS --
94180CPR      NAME  ATTRIBUTES      DEFINITION
94190CPR
94200CPR      CMNMX      (I)          ARRAY WITH MIN AND MAX WINDOW COORDS
94210CPR      CRD        (I)          ARRAY WITH ELM VERTICE COORDS
94220CPR      XP          (I) >>
94230CPR      YP          (I) >>>> STRESS POINT COORDS
94240CPR      ZP          (I) >>
94250CPR      NSP          (I)          STRESS POINT NAME
94260CPR      XLLOC        (O) >>
94270CPR      YLOC        (O) >>>> LOCAL COORDS OF STRESS POINT
94280CPR      ZLOC        (O) >>
94290CPR      SF          (O)          WEIGHTING COEFFICIENTS
94300CPR      ICONV        (O)          CONVERGENCE PARAMETER
94310CPR      0 --- DID NOT CONVERGE
94320CPR      1 --- CONVERGED TO POINT IN THE ELM
94330CPR      2 --- CONVERGED TO POINT OUT OF ELM
94340CPR      IPAS          (I)          SEARCH TYPE PARAMETER
94350CPR      -1 --- NO REARRANGE,NO ROTATION
94360CPR      NO SURFACING
94370CPR      0 --- NO REARRANGE,ROTATE
94380CPR      NO SURFACING
94390CPR      1 --- REAARANGE,ROTATE
94400CPR      PERFORM SURFACING
94410CPR      DIST          (O)          DISTANCE FORM OUTSIDE POINT TO SURFACE
94420CPR      XSUR          (O) >>
94430CPR      YSUR          (O) >>>> SURFACE COORDINATES
94440CPR      ZSUR          (O) >>
94450CPR
94460CPR      COMMONS USED --
94470CPR
94480CPR      AFIL
94490CPR
94500CPR      FUNCTIONS/ROUTINES CALLED --
94510CPR
94520CPR      REAARY
94530CPR      SHAPFV
94540CPR      CHKPT
94550CPR      JACBCK
94560CPR      SFDIST
94570CPR
94580CPR      FILES USED --
94590CPR
94600CPR
94610CPR      LIBRARIES ACCESSED --

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=88

```
9462OCPR  NONE
9463OCPR
9464OCPR
9465OCPR  LOCAL VARIABLES --
9466OCPR      IPOINT --- ACCURACY INDICATOR
9467OCPR          0 -- STRESS POINT NOT RECALCULATED ACCURATELY
9468OCPR          1 -- STRESS POINT      RECALCULATED ACCURATELY
9469OCPR      IJACB --- JACOBIAN CHECK PARAMETER
9470OCPR          0 -- JACOBIAN NOT POSITIVE
9471OCPR          1 -- JACOBIAN IS  POSITIVE
9472OCPR
9473OCPR  SPECIAL COMMENTS ABOUT THIS ROUTINE
9474OCPR
9475OCPR
9476OCPR  SPECIAL REMARKS/INSTRUCTIONS --
9477OCPR      HOST = H6000
9478OCPR
9479OCPR
9480OCPR=====
97960    END
```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=89

```

97980C=====
98000      SUBROUTINE REAARY(X,Y,Z,XP,YP,ZP,SF,ICLOS,ITRYC,DTOCN,XL,YL,ZL
98020CPR
98030CPR=====
98040CPR
98050CPR
98060CPR      ROUTINE--          REAARY
98070CPR
98080CPR      DATE/WRITTEN BY-- 11/28/83      R J  MAFFEO
98090CPR      DATE/REVISED BY--
98100CPR
98110CPR      FUNCTION/PURPOSE--
98120CPR
98130CPR      THIS ROUTINE REARRANGES THE CONNECTIVITY COORDS AND
98140CPR      THE SHAPE FUNCTION ARRAYS IN AN ATTEMPT TO GET AN
98150CPR      ORDER THAT WILL ENHANCE CONVERGENCE OF THE INVERSE
98160CPR      SHAPE FUNCTION ROUTINE(I3DSF) IT ALSO COMPUTES THE
98170CPR      DISTANCE FROM THE STRESS POINT TO THE NEAREST ELM
98180CPR      CORNER OPTIONALLY IT WILL "UN-REARRANGE" THE
98190CPR      SHAPE FUNCTION ARRAY AND THE LOCAL COORDS
98200CPR
98210CPR      CALLING ARGUMENTS --
98220CPR      NAME          ATTRIBUTES      DEFINITION
98230CPR
98240CPR      X          (I/O) >>
98250CPR      Y          (I/O) >>>>      CONNECTIVITY COORD ARRAYS
98260CPR      Z          (I/O) >>
98270CPR      XP         (I)  >>
98280CPR      YP         (I)  >>>>      COORDS OF STRESS POINTS
98290CPR      ZP         (I)  >>
98300CPR      SF         (I/O)      SHAPE FUNCTION ARRAY
98310CPR      ICLOS      (I/O)      VALUE OF NEW I JOINT
98320CPR      ITRYC      (I)        CONTROL ARGUMENT
98330CPR      -1 -- REARRANGE SHAPE FUNCTION ARRAY
98340CPR      0 -- PICK I JOINT AS JOINT NEAREST TO
98350CPR      STRESS POINT
98360CPR      N>0 -- USE VALUE OF N AS I JOINT
98370CPR      DISTANCE TO NEAREST CORNER
98380CPR      XL         (I/O) >>>
98390CPR      YL         (I/O) >>>>>      VALUES OF LOCAL COORDINATES
98400CPR      ZL         (I/O) >>>
98410CPR      COMMONS USED --
98420CPR
98430CPR
98440CPR      FUNCTIONS/ROUTINES CALLED --
98450CPR
98460CPR      SORT2
98470CPR
98480CPR      FILES USED --
98490CPR
98500CPR
98510CPR      LIBRARIES ACCESSED --
98520CPR      NONE
98530CPR
98540CPR
98550CPR      LOCAL VARIABLES --
98560CPR
98570CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
98580CPR
98590CPR      NOTE THE IJ ARRAY IS USED TO RELATE THE REARRANGED
98600CPR      CONNECTIVITY ARRAY TO THE JOINT PICKED AS THE I JOINT
98610CPR
99530      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=90

```
99550C=====
99570      SUBROUTINE SORT2(SEEDS,FOLLO,LAX)
99590C=====
99610CPR    ROUTINE--          SORT2
99620CPR
99630CPR    DATE/WRITTEN BY-- ??????????      GE DOUBLE SORT
99640CPR    DATE/REVISED BY--
99650CPR
99660CPR    FUNCTION/PURPOSE--
99670CPR
99680CPR    THIS ROUTINE SORTS TWO 1 DIMENSIONAL ARRAYS
99690CPR
99700CPR    CALLING ARGUMENTS --
99710CPR      NAME      ATTRIBUTES      DEFINITION
99720CPR
99730CPR      SEEDS      (I/O)      PRIMARY ARRAY TO BE SORTED
99740CPR      FOLLO      (I/O)      ARRAY TO BE SORTED LIKE THE PRIMARY
99750CPR      LAX        (I)        NUMBER OF ITEMS IN ARRAYS
99760CPR
99770CPR    COMMONS USED -- NONE
99780CPR
99790CPR    FUNCTIONS/ROUTINES CALLED -- NONE
99800CPR
99810CPR    FILES USED -- NONE
99820CPR
99830CPR    LIBRARIES ACCESSED --
99840CPR    NONE
99850CPR
99860CPR    LOCAL VARIABLES --
99870CPR
99880CPR    SPECIAL COMMENTS ABOUT THIS ROUTINE
99890CPR
99900CPR
99910CPR    SPECIAL REMARKS/INSTRUCTIONS --
99920CPR      HOST = H6000
99930CPR
99940CPR
99950CPR=====
100260      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=91

```

100280C=====
100300      SUBROUTINE SFDIST(SF,DIST,XNC,YNC,ZNC,XSUR,YSUR,ZSUR,XP,YP,ZP,
100330C=====
100340CPR
100350CPR=====
100360CPR
100370CPR
100380CPR      ROUTINE--          SFDIST
100390CPR
100400CPR      DATE/WRITTEN BY-- 11/15/83      RJ MAFFEO
100410CPR      DATE/REVISED BY--
100420CPR
100430CPR      FUNCTION/PURPOSE--
100440CPR
100450CPR      THIS ROUTINE COMPUTES THE DISTANCE FROM STRESS
100460CPR      POINTS THAT ARE OUTSIDE A HEAT TRANSFER ELM TO THE
100470CPR      SURFACE OF THE HEAT TRAN ELM IT ALSO COMPUTES THE
100480CPR      SURFACE COORDIANTES THIS SURFACE POINT IS FOUND BY
100490CPR      'TRUNCATING' THE LOCAL COORDINATES TO 1 OR -1
100500CPR
100510CPR
100520CPR
100530CPR      CALLING ARGUMENTS --
100540CPR          NAME          ATTRIBUTES          DEFINITION
100550CPR
100560CPR          SF          (I)          SHAPE FUNCTION ARRAY
100570CPR          DIST        (O)          COMPUTED DISTANCE
100580CPR          XNC          (I) >>>
100590CPR          YNC          (I) >>>>>>>> LOCAL VALUES OF STRESS POINT
100600CPR          ZNC          (I) >>>
100610CPR          XSUR        (O) >>>
100620CPR          YSUR        (O) >>>>>>>> COORDS OF SURFACE POINT
100630CPR          ZSUR        (O) >>>
100640CPR          XP          (I) >>>
100650CPR          YP          (I) >>>>>>>> COORDS OF STRESS POINT
100660CPR          ZP          (I) >>>
100670CPR          NNPE        (I)          NUMBER OF NODES PER ELM
100680CPR          CRD          (I)          COORDS OF VERTICES OF ELM
100690CPR
100700CPR      COMMONS USED --
100710CPR
100720CPR
100730CPR      FUNCTIONS/ROUTINES CALLED --
100740CPR
100750CPR      SHAPFV
100760CPR
100770CPR      FILES USED --
100780CPR
100790CPR
100800CPR      LIBRARIES ACCESSED --
100810CPR      NONE
100820CPR
100830CPR
100840CPR      LOCAL VARIABLES --
100850CPR
100860CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
100870CPR
100880CPR
100890CPR      SPECIAL REMARKS/INSTRUCTIONS --
100900CPR          HOST = H6000
100910CPR
100920CPR
100930CPR=====
101290      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=92

```

101310C=====
101330      SUBROUTINE CHKPT(SF,X,Y,Z,XS,YS,ZS,IPOINT)
101340CPR
101350CPR=====
101360CPR
101370CPR
101380CPR      ROUTINE--          CHKPT
101390CPR
101400CPR      DATE/WRITTEN BY-- 11/28/83      R J. MAFFEO
101410CPR      DATE/REVISED BY--
101420CPR
101430CPR      FUNCTION/PURPOSE--
101440CPR      THIS ROUTINE RECOMPUTES THE STRESS POINT COORDS BASED
101450CPR      ON THE VALUES OF THE LOCAL COORDINATES TO CHECK FOR
101460CPR      ACCURACY
101470CPR
101480CPR      CALLING ARGUMENTS --
101490CPR          NAME          ATTRIBUTES      DEFINITION
101500CPR
101510CPR          SF              (I)              ARRAY OF SHAPE FUNCTIONS
101520CPR          X              (I) >>
101530CPR          Y              (I) >>>> ARRAY OF ELM CORNER COORDS
101540CPR          Z              (I) >>
101550CPR          XS              (I) >>
101560CPR          YS              (I) >>>> ORIGINAL STRESS POINT COORDS
101570CPR          ZS              (I) >>
101580CPR          IPOINT          (0)              ACCURACY INDICATOR
101590CPR                                     0 -- DID NOT REPRODUCE COORD
101600CPR                                     1 -- DID      REPRODUCE COORD
101610CPR
101620CPR      COMMONS USED --
101630CPR
101640CPR
101650CPR      FUNCTIONS/ROUTINES CALLED --
101660CPR
101670CPR
101680CPR      FILES USED --
101690CPR
101700CPR
101710CPR      LIBRARIES ACCESSED --
101720CPR      NONE
101730CPR
101740CPR
101750CPR      LOCAL VARIABLES --
101760CPR
101770CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
101780CPR
101790CPR
101800CPR      SPECIAL REMARKS/INSTRUCTIONS --
101810CPR          HOST = H6000
101820CPR
101830CPR
101840CPR=====
101870C=====
102110      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=93

```

102130C=====
102150      SUBROUTINE PDOFSF(TETA,YNETA,ZETA,IGP,IND,PD)
102170C=====
102190CPR
102200CPR=====
102210CPR
102220CPR
102230CPR      ROUTINE--          PDOFSF
102240CPR
102250CPR      DATE/WRITTEN BY-- 09/14/82      RJ MAFFEO
102260CPR      DATE/REVISED BY--
102270CPR
102280CPR      FUNCTION/PURPOSE--
102290CPR
102300CPR      THIS SUB COMPUTES THE DERIVATIVES OF THE
102310CPR      SHAPE FUNCTIONS WRT THE LOCAL SYSTEM
102320CPR
102330CPR      CALLING ARGUMENTS --
102340CPR          NAME          ATTRIBUTES      DEFINITION
102350CPR
102360CPR          TETA          (I) >
102370CPR          YNETA        (I) >>          LOCAL COORDINATES
102380CPR          ZETA          (I) >
102390CPR          IGP            (I)              GAUSS POINT COUNTER
102400CPR          IND            (I)              2D/3D INDICATOR
102410CPR                                     IND=0 >> 3D
102420CPR                                     IND=1 THRU 6 >> 2D IND=FACE NUMBER
102430CPR          PD            (0)              ARRAY OF DERIVATIVES
102440CPR
102450CPR      COMMONS USED --
102460CPR
102470CPR          CORNDEF
102480CPR
102490CPR      FUNCTIONS/ROUTINES CALLED --
102500CPR
102510CPR
102520CPR      FILES USED --
102530CPR
102540CPR
102550CPR      LIBRARIES ACCESSED --
102560CPR      NONE
102570CPR
102580CPR
102590CPR      LOCAL VARIABLES --
102600CPR
102610CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
102620CPR
102630CPR
102640CPR      SPECIAL REMARKS/INSTRUCTIONS --
102650CPR          HOST = H6000
102660CPR
102670CPR
102680CPR=====
103040      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=94

```
103060C=====
103080      SUBROUTINE FNDJAC(X,Y,Z,P3D,XJAC)
103090CPR
103100CPR=====
103110CPR
103120CPR
103130CPR      ROUTINE--          FNDJAC
103140CPR
103150CPR      DATE/WRITTEN BY-- 11/28/83      R.J  MAFFEO
103160CPR      DATE/REVISED BY--
103170CPR
103180CPR      FUNCTION/PURPOSE--
103190CPR      THIS ROUTINE COMPUTES THE VALUE OF THE DETERMINANT OF
103200CPR      THE JACOBIAN EVALUATED AT THE STRESS POINT
103210CPR
103220CPR      CALLING ARGUMENTS --
103230CPR          NAME          ATTRIBUTES      DEFINITION
103240CPR
103250CPR          X          (I) >>
103260CPR          Y          (I) >>>> ARRAY OF ELM CORNER COORDS
103270CPR          Z          (I) >>
103280CPR          P3D         (I)          ARRAY WITH PARTIAL DERIVATIVES OF
103290CPR                                LOCAL COORD WRT THE GLOBAL COORD
103300CPR          XJAC      (O)          VALUE OF DETERMINANT OF JACOBIAN
103310CPR
103320CPR      COMMONS USED --
103330CPR
103340CPR
103350CPR      FUNCTIONS/ROUTINES CALLED --
103360CPR
103370CPR
103380CPR      FILES USED --
103390CPR
103400CPR
103410CPR      LIBRARIES ACCESSED --
103420CPR      NONE
103430CPR
103440CPR
103450CPR      LOCAL VARIABLES --
103460CPR
103470CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
103480CPR
103490CPR
103500CPR      SPECIAL REMARKS/INSTRUCTIONS --
103510CPR          HOST = H6000
103520CPR
103530CPR
103540CPR=====
103570C=====
103810      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=95

```
103830C=====
103850      SUBROUTINE SHAPFV(SF,XNC,YNC,ZNC)
103860CPR
103870CPR=====
103880CPR
103890CPR
103900CPR      ROUTINE--          SHAPFV
103910CPR
103920CPR      DATE/WRITTEN BY-- 11/28/83      R J  MAFFEO
103930CPR      DATE/REVISED BY--
103940CPR
103950CPR      FUNCTION/PURPOSE--
103960CPR      THIS ROUTINE FINDS THE VALUES OF THE SHAPE FUNCTIONS
103970CPR      FOR THE 8-NODED ISOPARAMETRIC ELM BASED ON THE VALUES
103980CPR      OF THE LOCAL COORDINATES
103990CPR
104000CPR      CALLING ARGUMENTS --
104010CPR          NAME          ATTRIBUTES      DEFINITION
104020CPR
104030CPR          SF          (O)          ARRAY WITH SHAPE FUNCTIONS
104040CPR          XNC          (I) >>
104050CPR          YNC          (I) >>>>      LOCAL COORDS
104060CPR          ZNC          (I) >>
104070CPR
104080CPR      COMMONS USED --
104090CPR
104100CPR
104110CPR      FUNCTIONS/ROUTINES CALLED --
104120CPR
104130CPR
104140CPR      FILES USED --
104150CPR
104160CPR
104170CPR      LIBRARIES ACCESSED --
104180CPR      NONE
104190CPR
104200CPR      LOCAL VARIABLES --
104210CPR
104220CPR
104230CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
104240CPR
104250CPR
104260CPR      SPECIAL REMARKS/INSTRUCTIONS --
104270CPR          HOST = H6000
104280CPR
104290CPR
104300CPR=====
104330CPR=====
104490      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=96

```

104510C=====
104530      SUBROUTINE JACBCK(XL,YL,ZL,X,Y,Z,XJAC,IJACB)
104540CPR
104550CPR=====
104560CPR
104570CPR
104580CPR      ROUTINE--          JACBCK
104590CPR
104600CPR      DATE/WRITTEN BY-- 11/28/83      R.J  MAFFEO
104610CPR      DATE/REVISED BY--
104620CPR
104630CPR      FUNCTION/PURPOSE--
104640CPR      THIS ROUTINE CALLS THE REQUIRED ROUTINES TO CHECK THE
104650CPR      VALUE OF THE DETERMINANT OF THE JACOBIAN AT THE STRESS
104660CPR      POINT
104670CPR
104680CPR      CALLING ARGUMENTS --
104690CPR          NAME          ATTRIBUTES      DEFINITION
104700CPR
104710CPR          XL          (I) >>
104720CPR          YL          (I) >>>>      LOCAL COORDS
104730CPR          ZL          (I) >>
104740CPR          X            (I) >>
104750CPR          Y            (I) >>>>      ARRAY OF ELM CORNER COORDS
104760CPR          Z            (I) >>
104770CPR          XJAC         (O)              VALUE OF DETERMINANT OF JACOBIAN
104780CPR          IJACB        (O)              INTERGER VARIABLE
104790CPR                                      0 -- JACOBIAN NOT POSITIVE
104800CPR                                      1 -- JACOBIAN IS POSITIVE
104810CPR
104820CPR      COMMONS USED --
104830CPR
104840CPR
104850CPR      FUNCTIONS/ROUTINES CALLED --
104860CPR
104870CPR          PDOFSF
104880CPR          FNDJAC
104890CPR
104900CPR      FILES USED --
104910CPR
104920CPR
104930CPR      LIBRARIES ACCESSED --
104940CPR      NONE
104950CPR
104960CPR
104970CPR      LOCAL VARIABLES --
104980CPR
104990CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
105000CPR
105010CPR
105020CPR      SPECIAL REMARKS/INSTRUCTIONS --
105030CPR          HOST = H6000
105040CPR
105050CPR
105060CPR=====
105090C=====
105250      END

```


TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=97

```

105270C=====
105290      SUBROUTINE RWSPCF(IREC,B,IRW)
105310C=====
105330CPR
105340CPR=====
105350CPR
105360CPR
105370CPR      ROUTINE--          RWSPCF
105380CPR
105390CPR      DATE/WRITTEN BY-- 04/07/83      RJ MAFFEO
105400CPR      DATE/REVISED BY--
105410CPR
105420CPR      FUNCTION/PURPOSE--
105430CPR
105440CPR      THIS ROUTINE READS AND WRITES DATA TO THE
105450CPR      THE RANDOM STRESS POINT COEFF FILE
105460CPR      THIS FILE CONTAINS THE FOLLOWING INFO PER RECORD
105470CPR      - STRESS NODE NAME      (1)
105480CPR      - HEAT TRAN ELM CONTAINING THIS STRESS POINT  (1)
105490CPR      - CONNECTIVITY OF THE HEAT TRAN ELM  (8)
105500CPR      - WEIGHTING COEFF FOR THIS STRESS NODE (8)
105510CPR
105520CPR
105530CPR
105540CPR      CALLING ARGUMENTS --
105550CPR      NAME      ATTRIBUTES      DEFINITION
105560CPR
105570CPR      IREC      (I)      FILE RECORD NUMBER
105580CPR      B        (I/O)     DATA ARRAY
105590CPR      IRW      (I)      READ/WRITE OPTION
105600CPR                        (0-READ 1-WRITE)
105610CPR
105620CPR      COMMONS USED --
105630CPR
105640CPR      AFIL
105650CPR
105660CPR      FUNCTIONS/ROUTINES CALLED --
105670CPR
105680CPR
105690CPR      FILES USED --
105700CPR
105710CPR      IRSPCF  R
105720CPR
105730CPR      LIBRARIES ACCESSED --
105740CPR      NONE
105750CPR
105760CPR
105770CPR      LOCAL VARIABLES --
105780CPR
105790CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
105800CPR
105810CPR
105820CPR      SPECIAL REMARKS/INSTRUCTIONS --
105830CPR      HOST = H6000
105840CPR
105850CPR
105860CPR=====
106170      END

```

TITLE= DATE =07/15/85 TIME = 23 42 PAGE=98

```
106190C=====
106210      SUBROUTINE ELTRAN
106230C=====
106250CPR
106260CPR=====
106270CPR
106280CPR
106290CPR      ROUTINE--          ELTRAN
106300CPR
106310CPR      DATE/WRITTEN BY-- 01/27/84      RJ MAFFEO
106320CPR      DATE/REVISED BY--
106330CPR
106340CPR      FUNCTION/PURPOSE--
106350CPR
106360CPR      Validate the values input for IECODE and IACODE then assign
106370CPR      the values to NNPE and NTPE depending on the values of
106380CPR      IECODE and IACODE
106390CPR
106400CPR
106410CPR
106420CPR
106430CPR      CALLING ARGUMENTS --
106440CPR      NAME          ATTRIBUTES      DEFINITION
106450CPR
106460CPR
106470CPR      COMMONS USED --
106480CPR
106490CPR      AFIL
106500CPR      CNTLFL
106510CPR
106520CPR      FUNCTIONS/ROUTINES CALLED --
106530CPR
106540CPR
106550CPR      FILES USED --
106560CPR
106570CPR      IERF      S
106580CPR
106590CPR      LIBRARIES ACCESSED --
106600CPR      NONE
106610CPR
106620CPR
106630CPR      LOCAL VARIABLES --
106640CPR
106650CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
106660CPR
106670CPR
106680CPR      SPECIAL REMARKS/INSTRUCTIONS --
106690CPR      HOST = H6000
106700CPR
106710CPR
106720CPR=====
107330      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=99

```
107350C=====
107370      SUBROUTINE INITCM
107390C=====
107410CPR
107420CPR=====
107430CPR
107440CPR
107450CPR      ROUTINE--          INITCM
107460CPR
107470CPR      DATE/WRITTEN BY-- 01/31/84      RJ MAFFEO
107480CPR      DATE/REVISED BY--
107490CPR
107500CPR      FUNCTION/PURPOSE--
107510CPR
107520CPR      THIS ROUTINE INITIALIZES THE CRNDEF COMMON
107530CPR      THIS COMMON IS USED IN CALCRD AND I3DSF ROUTINES
107540CPR
107550CPR      CALLING ARGUMENTS --
107560CPR          NAME          ATTRIBUTES      DEFINITION
107570CPR
107580CPR
107590CPR      COMMONS USED --
107600CPR
107610CPR          CRNDEF
107620CPR
107630CPR      FUNCTIONS/ROUTINES CALLED --
107640CPR
107650CPR
107660CPR      FILES USED --
107670CPR
107680CPR
107690CPR      LIBRARIES ACCESSED --
107700CPR      NONE
107710CPR
107720CPR
107730CPR      LOCAL VARIABLES --
107740CPR
107750CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
107760CPR
107770CPR
107780CPR      SPECIAL REMARKS/INSTRUCTIONS --
107790CPR          HOST = H6000
107800CPR
107810CPR
107820CPR=====
108390      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=**

```

108410C=====
108430      SUBROUTINE MARCTO(ISPOC,NUMSP,NUMSOL)
108450C=====
108470CPR
108480CPR=====
108490CPR
108500CPR
108510CPR      ROUTINE--          MARCTO
108520CPR
108530CPR      DATE/WRITTEN BY-- 03/08/84      RJ MAFFEO
108540CPR      DATE/REVISED BY--
108550CPR
108560CPR      FUNCTION/PURPOSE--
108570CPR
108580CPR      THIS ROUTINE CALLS ALL REQUIRED ROUTINES TO COMPUTE
108590CPR      THE MARC 8-NODED ELM GAUSS POINT TEMPERATURES(OR THE
108600CPR      ELM CENTROIDAL TEMPERATURE)
108610CPR
108620CPR      CALLING ARGUMENTS --
108630CPR          NAME          ATTRIBUTES      DEFINITION
108640CPR
108650CPR          ISPOC      (I)    DESIRED ELM LOCATION
108660CPR                      ISPOC=2 --- CENTROIDAL TEMPERATURE
108670CPR                      ISPOC=3 --- 2X2X2 GAUSS POINT TEMPERATURES
108680CPR          NUMSP      (I)    NUMBER OF STRESS MODEL NODAL POINTS
108690CPR          NUMSOL     (I)    NUMBER OF TRANSIENT SOLUTIONS
108700CPR
108710CPR      COMMONS USED --
108720CPR
108730CPR          AFIL
108740CPR          WORK
108750CPR
108760CPR      FUNCTIONS/ROUTINES CALLED --
108770CPR
108780CPR          GPCDEF
108790CPR          GPTEMP
108800CPR
108810CPR      FILES USED --
108820CPR
108830CPR          ISNEF      S
108840CPR          IOUTF      S
108850CPR          ITEMP      S
108860CPR          IERF       S
108870CPR
108880CPR      LIBRARIES ACCESSED --
108890CPR      NONE
108900CPR
108910CPR
108920CPR      LOCAL VARIABLES --
108930CPR
108940CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
108950CPR
108960CPR
108970CPR      SPECIAL REMARKS/INSTRUCTIONS --
108980CPR          HOST = H6000
108990CPR
109000CPR
109010CPR=====
109950      END

```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=**

```
109970C=====
109990      SUBROUTINE GPCDEF(GPF)
110010C=====
110030CPR
110040CPR=====
110050CPR
110060CPR
110070CPR      ROUTINE--          GPCDEF
110080CPR
110090CPR      DATE/WRITTEN BY-- 03/07/84      RJ MAFFEO
110100CPR      DATE/REVISED BY--
110110CPR
110120CPR      FUNCTION/PURPOSE--
110130CPR
110140CPR      THIS ROUTINE COMPUTES AND STORES THE WEIGHTING
110150CPR      COEFFS ASSOCIATED WITH THE 2X2X2 GAUSS POINTS
110160CPR      USED FOR THE MARC SOLID ELM(ELM TYPE 43)
110170CPR
110180CPR
110190CPR
110200CPR      CALLING ARGUMENTS --
110210CPR          NAME          ATTRIBUTES      DEFINITION
110220CPR
110230CPR          GPF          (0)      ARRAY WITH GAUSS PT COEFF
110240CPR
110250CPR      COMMONS USED --
110260CPR
110270CPR
110280CPR      FUNCTIONS/ROUTINES CALLED --
110290CPR
110300CPR
110310CPR      FILES USED --
110320CPR
110330CPR
110340CPR      LIBRARIES ACCESSED --
110350CPR      NONE
110360CPR
110370CPR
110380CPR      LOCAL VARIABLES --
110390CPR
110400CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
110410CPR
110420CPR
110430CPR      SPECIAL REMARKS/INSTRUCTIONS --
110440CPR          HOST = H6000
110450CPR
110460CPR
110470CPR=====
110710      END
```

TITLE=

DATE =07/15/85

TIME = 23 42

PAGE=**

```

110730C=====
110750      SUBROUTINE GPTEMP(TA,IE,GPF,ISPOC,TV)
110770C=====
110790CPR
110800CPR=====
110810CPR
110820CPR
110830CPR      ROUTINE--          GPTEMP
110840CPR
110850CPR      DATE/WRITTEN BY-- 03/17/84      RJ MAFFEO
110860CPR      DATE/REVISED BY--
110870CPR
110880CPR      FUNCTION/PURPOSE--
110890CPR
110900CPR      THIS ROUTINE COMPUTES THE VALUE OF THE ELM TEMPERATURE
110910CPR      FOR THE 8-NODED 3D MARC STRESS ELM IT WILL COMPUTE
110920CPR      TEMPS AT THE 8 2X2X2 GAUSS POINTS OR AT THE ELM
110930CPR      CENTROID
110940CPR
110950CPR
110960CPR
110970CPR      CALLING ARGUMENTS --
110980CPR          NAME          ATTRIBUTES      DEFINITION
110990CPR
111000CPR          TA          (I)          NODAL TEMPERATURE ARRAY
111010CPR          IE          (I)          ELM CONNECTIVITY ARRAY
111020CPR          GPF          (I)          GAUSS PT COEFF ARRAY
111030CPR          ISPOC       (I)          TEMPERATURE LOCATION PARAMETER
111040CPR          TV          (O)          ELEMENTAL TEMPERATURE ARRAY
111050CPR                                     NOTE IF ISPOC=2
111060CPR                                     TV(1)=CENTROIDAL TEMPERATURE
111070CPR
111080CPR      COMMONS USED --
111090CPR
111100CPR
111110CPR      FUNCTIONS/ROUTINES CALLED --
111120CPR
111130CPR
111140CPR      FILES USED --
111150CPR
111160CPR
111170CPR      LIBRARIES ACCESSED --
111180CPR      NONE
111190CPR
111200CPR
111210CPR      LOCAL VARIABLES --
111220CPR
111230CPR      SPECIAL COMMENTS ABOUT THIS ROUTINE
111240CPR
111250CPR
111260CPR      SPECIAL REMARKS/INSTRUCTIONS --
111270CPR          HOST = H6000
111280CPR
111290CPR
111300CPR=====
111590      END

```

APPENDIX D

• Transfer Module Flow Diagram

This appendix contains a program flow diagram of the transfer module. This diagram shows the general flow of the code and which subroutines are subordinate to each other.

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 1

```
*****
*
*   THE SAVAM STACKER   *
*
*****
```

THIS IS CALLING STACK STARTING FROM \$\$\$ MAI

```
*****
MAIN
*****
```

```

/
/
/ *****
/ \\*INITCM *
/ *****
/      001
/
/
/ *****
/ \\*RFILCR *
/ *****
/      002
/
/
/ *****
/ //////////////*RANSIZ *
/ *****
/      L003
/
/
/ *****
/ //////////////*ZERORF *
/ *****
/      004
/
/
/ *****
/ \\*FTIME  *
/ *****
/      005
/
/
/ *****
/ \\*HTICON  *
/ *****
/      006
/
/
```

```

/
/
/ *****
/ //////////////*SINTIT *
/ *****
/
```


TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 2

007

*DATINT *

008

*PRCSOF *

009

*GETEWT *

010

*ERRPRT *

011

*GTNUMT *

012

*ERRPRT =

RO11

*PRCGEO *

013

*ERRPRT =

RO11

*WRTNTF *

014

*WRTDIR *

PAGE= 3

```

      /
     /
    /
   /
  /
 /
/
//
///
////
=====
=ERRPRT =
=====
RO11

*****
**WRTGED **
*****
   /
  /
 /
/
//
///
////
=====
=ERRPRT =
=====
RO11

*****
**WRTEMP **
*****
   /
  /
 /
/
//
///
////
=====
=ERRPRT =
=====
RO11

*****
**          **
*****
   /
  /
 /
/
//
///
////
=====
=ERRPRT =
=====
RO11
```

```

\ \ O15
\ \
\ \ =====
\ \ \ \ \ \ \ \ =ERRPRT =
\ \ \ \ \ \ \ \ =====
\ \ RO11

*****
GEO *
*****
\ \ O16
\ \
\ \ =====
\ \ \ \ \ \ \ \ =ERRPRT =
\ \ \ \ \ \ \ \ =====
\ \ RO11

*****
EMP *
*****
\ \ O17
\ \
\ \ =====
\ \ \ \ \ \ \ \ =ERRPRT =
\ \ \ \ \ \ \ \ =====
\ \ RO11

=====
RT =
=====
RO11
```

PAGE= 4

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 5

```

/
/      O30
/
/ *****
/ \\*RDTIME *
/ *****
/      O31
/
/      *****
/      \\*HTFIO *
/      *****
/      RO22
/
/      *****
/      \\*SKIP *
/      *****
/      O32
/
/      *****
/      \\*TIMCK *
/      *****
/      O33
/
/      *****
/      \\*CHNODT *
/      *****
/      O34
/
/
/
```

```

/
/
/      *****
/      \\*HTFIO *
/      *****
/      RO22
/
/      *****
/      \\*RWELMS *
/      *****
/      RO28
/
/ *****
/ \\*PREPGM *
/ *****
/      O35
/
/      *****
/      \\*RWELMS *
/      *****
/      RO28
/
/      *****
/      \\*CTRAN *
/      *****
/
/
/
```

PAGE= 6

```

*****
//*****GETPD*****
//*****
O40

*****
//*****PARD*****
//*****
O41

*****
//*****GTGAUS*****
//*****
O42

=====
//=====RWELMS=====
=====
R028

*****
//*****VQLM*****
//*****
O43

*****
//*****SHAPEF*****
//*****

```

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 7

O44

////////// *AREAD *

O45

////////// *GTGAUS *

RO42

////////// *SHAPEF *

RO44

////////// *REACRD *

O46

////////// *SSURCH *

O47

////////// *KNTSPF *

O48

////////// *FILREW *

O49

////////// *RDSPF *

O50

////////// *GNLSUR *

O51

////////// *EINCOR *

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 9

059

*CHKPT *

060

*JACBCK *

061

*PDOFSF *

062

*FNDJAC *

063

*SFDIST *

064

*SHAPFV *

R059

\\\\\\\\\\\\\\\\=RWELMS =

R028

*RWSPCF *

065

*CORNCC *

066

\\\\\\\\\\\\\\\\=RWELMS =

PAGE=10

$$\begin{array}{cc} \diagup & \diagup \\ \diagdown & \diagdown \end{array}$$

PAGE= 10

```

/      /          /
/      /          /
/      /          /=====
//     //         ///=FILREW =

```

PAGE= 12

```

\          ****
\\\\\\\\\\\\\\*NTCORE*

```

PAGE = 13

```

*****
*GPTEMP*
*****
092

```

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 1

MAIN	SU 1	MAIN		NEWSRC	100001
MAIN	CO 3	MAFCOM	M	NEWSRC	100001
MAIN	CO 3	AFIL	C	NEWSRC	100001
MAIN	CO 3	MAFCOM	M	NEWSRC	100001
MAIN	CO 3	ELMDAT	C	NEWSRC	100001
MAIN	CO 3	MAFCOM	M	NEWSRC	100001
MAIN	CO 3	SIZE	C	NEWSRC	100001
MAIN	CO 3	MAFCOM	M	NEWSRC	100001
MAIN	CO 3	WORK	C	NEWSRC	100001
MAIN	CO 3	MAFCOM	M	NEWSRC	100001
MAIN	CO 3	CNTLGM	C	NEWSRC	100001
MAIN	CO 3	MAFCOM	M	NEWSRC	100001
MAIN	CO 3	CNTLTM	C	NEWSRC	100001
MAIN	CO 3	MAFCOM	M	NEWSRC	100001
MAIN	CO 3	SPDAT	C	NEWSRC	100001
MAIN	CO 3	CNTLFL	C	NEWSRC	100001
MAIN	CO 3	MAFCOM	M	NEWSRC	100001
MAIN	CO 3	CRNDEF	C	NEWSRC	100001
MAIN	SU 5 101	INITCM		NEWSRC	100001
MAIN	SU 5 102	RFILCR		NEWSRC	100001
MAIN	SU 5 103	FTIME		NEWSRC	100001
MAIN	SU 5 104	HTICON		NEWSRC	100001
MAIN	SU 5 105	RDCNTL		NEWSRC	100001
MAIN	SU 5 106	MKRGF		NEWSRC	100001
MAIN	SU 5 108	PRNTIM		NEWSRC	100001
MAIN	SU 5 109	RDTIME		NEWSRC	100001
MAIN	SU 5 112	PREPGM		NEWSRC	100001
MAIN	SU 5 115	CALCRD		NEWSRC	100001
MAIN	SU 5 118	SSURCH		NEWSRC	100001
MAIN	SU 5 121	CORNCC		NEWSRC	100001
MAIN	SU 5 124	GENTMP		NEWSRC	100001
MAIN	SU 5 127	TEMFOR		NEWSRC	100001
MAIN	RD 7	YES		NEWSRC	100001
MAIN	WR 8	YES		NEWSRC	100001
RWNODS	SU 1	RWNODS		NEWSRC	100002
RWNODS	CO 3	MAFCOM	M	NEWSRC	100002
RWNODS	CO 3	AFIL	C	NEWSRC	100002
RWNODS	RD 7	YES		NEWSRC	100002
RWNODS	WR 8	YES		NEWSRC	100002
RWELMS	SU 1	RWELMS		NEWSRC	100003
RWELMS	CO 3	MAFCOM	M	NEWSRC	100003
RWELMS	CO 3	AFIL	C	NEWSRC	100003
RWELMS	RD 7	YES		NEWSRC	100003
RWELMS	WR 8	YES		NEWSRC	100003
RDTIME	SU 1	RDTIME		NEWSRC	100004
RDTIME	CO 3	MAFCOM	M	NEWSRC	100004
RDTIME	CO 3	AFIL	C	NEWSRC	100004
RDTIME	CO 3	MAFCOM	M	NEWSRC	100004
RDTIME	CO 3	SIZE	C	NEWSRC	100004
RDTIME	CO 3	MAFCOM	M	NEWSRC	100004
RDTIME	CO 3	CNTLTM	C	NEWSRC	100004
RDTIME	CO 3	CNTLFL	C	NEWSRC	100004
RDTIME	SU 5 101	HTFID		NEWSRC	100004
RDTIME	SU 5 102	SKIP		NEWSRC	100004
RDTIME	SU 5 104	TIMCK		NEWSRC	100004
RDTIME	SU 5 105	CHNODT		NEWSRC	100004
RDTIME	WR 8	YES		NEWSRC	100004
TIMCK	SU 1	TIMCK		NEWSRC	100005
TIMCK	CO 3	MAFCOM	M	NEWSRC	100005
TIMCK	CO 3	CNTLTM	C	NEWSRC	100005
CHNODT	SU 1	CHNODT		NEWSRC	100006
CHNODT	CO 3	MAFCOM	M	NEWSRC	100006
CHNODT	CO 3	AFIL	C	NEWSRC	100006
CHNODT	CO 3	CNTLFL	C	NEWSRC	100006
CHNODT	CO 3	MAFCOM	M	NEWSRC	100006
CHNODT	CO 3	ELMDAT	C	NEWSRC	100006
CHNODT	CO 3	MAFCOM	M	NEWSRC	100006

-- GE/Z09

--

PAGE= 1

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 2

CHNODT	CO 3	SIZE	C	NEWSRC	100006
CHNODT	CO 3	MAFCOM	M	NEWSRC	100006
CHNODT	CO 3	WORK	C	NEWSRC	100006
CHNODT	SU 5 101	HTFIO		NEWSRC	100006
CHNODT	SU 5 102	RWELMS		NEWSRC	100006
CHNODT	WR 8	YES		NEWSRC	100006
PREPGM	SU 1	PREPGM		NEWSRC	100007
PREPGM	CO 3	MAFCOM	M	NEWSRC	100007
PREPGM	CO 3	ELMDAT	C	NEWSRC	100007
PREPGM	CO 3	MAFCOM	M	NEWSRC	100007
PREPGM	CO 3	SIZE	C	NEWSRC	100007
PREPGM	CO 3	MAFCOM	M	NEWSRC	100007
PREPGM	CO 3	CNTLGM	C	NEWSRC	100007
PREPGM	SU 5 101	RWELMS		NEWSRC	100007
PREPGM	SU 5 102	CTTRAN		NEWSRC	100007
PREPGM	SU 5 103	SCTION		NEWSRC	100007
PREPGM	SU 5 105	REMELM		NEWSRC	100007
CTTRAN	SU 1	CTTRAN		NEWSRC	100008
SCTION	SU 1	SCTION		NEWSRC	100009
REMELM	SU 1	REMELM		NEWSRC	100010
REMELM	CO 3	MAFCOM	M	NEWSRC	100010
REMELM	CO 3	ELMDAT	C	NEWSRC	100010
REMELM	CO 3	MAFCOM	M	NEWSRC	100010
REMELM	CO 3	CNTLGM	C	NEWSRC	100010
REMELM	SU 5 101	RWELMS		NEWSRC	100010
CALCRD	SU 1	CALCRD		NEWSRC	100011
CALCRD	CO 3	MAFCOM	M	NEWSRC	100011
CALCRD	CO 3	AFIL	C	NEWSRC	100011
CALCRD	CO 3	MAFCOM	M	NEWSRC	100011
CALCRD	CO 3	ELMDAT	C	NEWSRC	100011
CALCRD	CO 3	MAFCOM	M	NEWSRC	100011
CALCRD	CO 3	SIZE	C	NEWSRC	100011
CALCRD	CO 3	MAFCOM	M	NEWSRC	100011
CALCRD	CO 3	WORK	C	NEWSRC	100011
CALCRD	CO 3	CRNDEF	C	NEWSRC	100011
CALCRD	SU 5 101	GETPD		NEWSRC	100011
CALCRD	SU 5 102	RWELMS		NEWSRC	100011
CALCRD	SU 5 103	VOLM		NEWSRC	100011
CALCRD	SU 5 104	AREAD		NEWSRC	100011
CALCRD	SU 5 105	REACRD		NEWSRC	100011
CALCRD	WR 8	YES		NEWSRC	100011
GETPD	SU 1	GETPD		NEWSRC	100012
GETPD	CO 3	CRNDEF	C	NEWSRC	100012
GETPD	SU 5 101	PARD		NEWSRC	100012
GETPD	SU 5 102	GTGAUS		NEWSRC	100012
VOLM	SU 1	VOLM		NEWSRC	100013
VOLM	SU 5 101	SHAPEF		NEWSRC	100013
SHAPEF	SU 1	SHAPEF		NEWSRC	100014
SHAPEF	CO 3	CRNDEF	C	NEWSRC	100014
PARD	SU 1	PARD		NEWSRC	100015
PARD	CO 3	CRNDEF	C	NEWSRC	100015
AREAD	SU 1	AREAD		NEWSRC	100016
AREAD	CO 3	CRNDEF	C	NEWSRC	100016
AREAD	SU 5 101	GTGAUS		NEWSRC	100016
AREAD	SU 5 102	SHAPEF		NEWSRC	100016
GTGAUS	SU 1	GTGAUS		NEWSRC	100017
REACRD	SU 1	REACRD		NEWSRC	100018
CORNCC	SU 1	CORNCC		NEWSRC	100019
CORNCC	CO 3	MAFCOM	M	NEWSRC	100019
CORNCC	CO 3	AFIL	C	NEWSRC	100019
CORNCC	CO 3	MAFCOM	M	NEWSRC	100019
CORNCC	CO 3	ELMDAT	C	NEWSRC	100019
CORNCC	CO 3	MAFCOM	M	NEWSRC	100019
CORNCC	CO 3	SIZE	C	NEWSRC	100019
CORNCC	CO 3	MAFCOM	M	NEWSRC	100019
CORNCC	CO 3	WORK	C	NEWSRC	100019
CORNCC	CO 3	MAFCOM	M	NEWSRC	100019

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 3

CORNCC	CO 3	CNTLGM	C	NEWSRC	100019
CORNCC	SU 5 101	RWELMS		NEWSRC	100019
CORNCC	SU 5 102	SORT		NEWSRC	100019
CORNCC	SU 5 103	CCNVEC		NEWSRC	100019
CORNCC	SU 5 105	GETCRD		NEWSRC	100019
CORNCC	SU 5 106	CINV		NEWSRC	100019
CORNCC	SU 5 108	RWCCF		NEWSRC	100019
CORNCC	SU 5 109	PUTWFI		NEWSRC	100019
CORNCC	WR 8	YES		NEWSRC	100019
CCNVEC	SU 1	CCNVEC		NEWSRC	100020
CCNVEC	CO 3	MAFCOM	M	NEWSRC	100020
CCNVEC	CO 3	AFIL	C	NEWSRC	100020
CCNVEC	WR 8	YES		NEWSRC	100020
GETCRD	SU 1	GETCRD		NEWSRC	100021
GETCRD	CO 3	MAFCOM	M	NEWSRC	100021
GETCRD	CO 3	AFIL	C	NEWSRC	100021
GETCRD	WR 8	YES		NEWSRC	100021
PUTWFI	SU 1	PUTWFI		NEWSRC	100022
PUTWFI	CO 3	MAFCOM	M	NEWSRC	100022
PUTWFI	CO 3	AFIL	C	NEWSRC	100022
RWCCF	SU 1	RWCCF		NEWSRC	100023
RWCCF	CO 3	MAFCOM	M	NEWSRC	100023
RWCCF	CO 3	AFIL	C	NEWSRC	100023
RWCCF	RD 7	YES		NEWSRC	100023
RWCCF	WR 8	YES		NEWSRC	100023
GETMPE	SU 1	GETMPE		NEWSRC	100024
GETMPE	CO 3	MAFCOM	M	NEWSRC	100024
GETMPE	CO 3	AFIL	C	NEWSRC	100024
GETMPE	CO 3	CNTLFL	C	NEWSRC	100024
GETMPE	CO 3	MAFCOM	M	NEWSRC	100024
GETMPE	CO 3	SIZE	C	NEWSRC	100024
GETMPE	SU 5 101	HTFIO		NEWSRC	100024
GETMPE	SU 5 103	MPERW		NEWSRC	100024
MPERW	SU 1	MPERW		NEWSRC	100025
MPERW	WR 8	YES		NEWSRC	100025
SETFLP	SU 1	SETFLP		NEWSRC	100026
SETFLP	CO 3	MAFCOM	M	NEWSRC	100026
SETFLP	CO 3	AFIL	C	NEWSRC	100026
SETFLP	SU 5 101	TFLSET		NEWSRC	100026
SETFLP	SU 5 102	HTFPAS		NEWSRC	100026
RDTEMP	SU 1	RDTEMP		NEWSRC	100027
RDTEMP	CO 3	MAFCOM	M	NEWSRC	100027
RDTEMP	CO 3	AFIL	C	NEWSRC	100027
RDTEMP	CO 3	MAFCOM	M	NEWSRC	100027
RDTEMP	CO 3	SIZE	C	NEWSRC	100027
RDTEMP	CO 3	MAFCOM	M	NEWSRC	100027
RDTEMP	CO 3	CNTLFL	C	NEWSRC	100027
RDTEMP	SU 5 101	HTFIO		NEWSRC	100027
RDTEMP	WR 8	YES		NEWSRC	100027
CWCRNT	SU 1	CWCRNT		NEWSRC	100028
CWCRNT	CO 3	MAFCOM	M	NEWSRC	100028
CWCRNT	CO 3	AFIL	C	NEWSRC	100028
CWCRNT	SU 5 101	FILREW		NEWSRC	100028
CWCRNT	SU 5 102	RWCCF		NEWSRC	100028
CWCRNT	SU 5 103	UNPAKF		NEWSRC	100028
CWCRNT	SU 5 104	MPERW		NEWSRC	100028
CWCRNT	SU 5 105	FCRNTP		NEWSRC	100028
CWCRNT	SU 5 106	MAXTDF		NEWSRC	100028
CWCRNT	SU 5 107	WRCRNT		NEWSRC	100028
UNPAKF	SU 1	UNPAKF		NEWSRC	100029
UNPAKF	CO 3	MAFCOM	M	NEWSRC	100029
UNPAKF	CO 3	AFIL	C	NEWSRC	100029
UNPAKF	WR 8	YES		NEWSRC	100029
FCRNTP	SU 1	FCRNTP		NEWSRC	100030
FCRNTP	CO 3	MAFCOM	M	NEWSRC	100030
FCRNTP	CO 3	AFIL	C	NEWSRC	100030
MAXTDF	SU 1	MAXTDF		NEWSRC	100031

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 4

MAXTDF	CO 3	MAFCOM	M	NEWSRC	100031
MAXTDF	CO 3	AFIL	C	NEWSRC	100031
MAXTDF	SU 5 101	SORT		NEWSRC	100031
MAXTDF	WR 8	YES		NEWSRC	100031
WRITMP	SU 1	WRITMP		NEWSRC	100032
WRITMP	CO 3	MAFCOM	M	NEWSRC	100032
WRITMP	CO 3	AFIL	C	NEWSRC	100032
WRITMP	WR 8	YES		NEWSRC	100032
CINV	SU 1	CINV		NEWSRC	100033
CINV	CO 3	MAFCOM	M	NEWSRC	100033
CINV	CO 3	AFIL	C	NEWSRC	100033
CINV	WR 8	YES		NEWSRC	100033
PRNTIM	SU 1	PRNTIM		NEWSRC	100034
PRNTIM	CO 3	MAFCOM	M	NEWSRC	100034
PRNTIM	CO 3	AFIL	C	NEWSRC	100034
PRNTIM	WR 8	YES		NEWSRC	100034
EMINMX	SU 1	EMINMX		NEWSRC	100035
EMINMX	SU 5 101	SORT		NEWSRC	100035
FTIME	SU 1	FTIME		NEWSRC	100036
SSURCH	SU 1	SSURCH		NEWSRC	100037
SSURCH	CO 3	MAFCOM	M	NEWSRC	100037
SSURCH	CO 3	AFIL	C	NEWSRC	100037
SSURCH	CO 3	MAFCOM	M	NEWSRC	100037
SSURCH	CO 3	SIZE	C	NEWSRC	100037
SSURCH	CO 3	MAFCOM	M	NEWSRC	100037
SSURCH	CO 3	SPDAT	C	NEWSRC	100037
SSURCH	SU 5 101	KNTSPF		NEWSRC	100037
SSURCH	SU 5 102	GNLSUR		NEWSRC	100037
KNTSPF	SU 1	KNTSPF		NEWSRC	100038
KNTSPF	CO 3	MAFCOM	M	NEWSRC	100038
KNTSPF	CO 3	AFIL	C	NEWSRC	100038
KNTSPF	SU 5 101	FILREW		NEWSRC	100038
KNTSPF	SU 5 102	RDSPF		NEWSRC	100038
KNTSPF	WR 8	YES		NEWSRC	100038
RFILCR	SU 1	RFILCR		NEWSRC	100039
RFILCR	CO 3	MAFCOM	M	NEWSRC	100039
RFILCR	CO 3	ELMDAT	C	NEWSRC	100039
RFILCR	SU 5 101	RANSIZ		NEWSRC	100039
RFILCR	SU 5 102	ZERORF		NEWSRC	100039
ZERORF	SU 1	ZERORF		NEWSRC	100040
ZERORF	WR 8	YES		NEWSRC	100040
GNLSUR	SU 1	GNLSUR		NEWSRC	100041
GNLSUR	CO 3	MAFCOM	M	NEWSRC	100041
GNLSUR	CO 3	AFIL	C	NEWSRC	100041
GNLSUR	CO 3	MAFCOM	M	NEWSRC	100041
GNLSUR	CO 3	SIZE	C	NEWSRC	100041
GNLSUR	CO 3	MAFCOM	M	NEWSRC	100041
GNLSUR	CO 3	WORK	C	NEWSRC	100041
GNLSUR	SU 5 101	EINCOR		NEWSRC	100041
GNLSUR	SU 5 102	RDSPF		NEWSRC	100041
GNLSUR	SU 5 103	FNDELM		NEWSRC	100041
GNLSUR	WR 8	YES		NEWSRC	100041
RDSPF	SU 1	RDSPF		NEWSRC	100042
RDSPF	CO 3	MAFCOM	M	NEWSRC	100042
RDSPF	CO 3	AFIL	C	NEWSRC	100042
RDSPF	RD 7	YES		NEWSRC	100042
FILREW	SU 1	FILREW		NEWSRC	100043
EINCOR	SU 1	EINCOR		NEWSRC	100044
EINCOR	CO 3	MAFCOM	M	NEWSRC	100044
EINCOR	CO 3	AFIL	C	NEWSRC	100044
EINCOR	CO 3	MAFCOM	M	NEWSRC	100044
EINCOR	CO 3	ELMDAT	C	NEWSRC	100044
EINCOR	SU 5 101	RWELMS		NEWSRC	100044
FNDELM	SU 1	FNDELM		NEWSRC	100045
FNDELM	CO 3	MAFCOM	M	NEWSRC	100045
FNDELM	CO 3	AFIL	C	NEWSRC	100045
FNDELM	CO 3	MAFCOM	M	NEWSRC	100045

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 5

FNDELM	CO 3	ELMDAT	C	NEWSRC	100045
FNDELM	CO 3	MAFCOM	M	NEWSRC	100045
FNDELM	CO 3	SIZE	C	NEWSRC	100045
FNDELM	CO 3	MAFCOM	M	NEWSRC	100045
FNDELM	CO 3	SPDAT	C	NEWSRC	100045
FNDELM	SU 5	101 GETWND		NEWSRC	100045
FNDELM	SU 5	102 CKWIND		NEWSRC	100045
FNDELM	SU 5	104 I3DSF		NEWSRC	100045
FNDELM	SU 5	105 RWELMS		NEWSRC	100045
FNDELM	SU 5	109 RWSPCF		NEWSRC	100045
FNDELM	WR 8	YES		NEWSRC	100045
GETWND	SU 1	GETWND		NEWSRC	100046
GETWND	CO 3	MAFCOM	M	NEWSRC	100046
GETWND	CO 3	AFIL	C	NEWSRC	100046
GETWND	CO 3	MAFCOM	M	NEWSRC	100046
GETWND	CO 3	ELMDAT	C	NEWSRC	100046
GETWND	CO 3	MAFCOM	M	NEWSRC	100046
GETWND	CO 3	SIZE	C	NEWSRC	100046
GETWND	CO 3	MAFCOM	M	NEWSRC	100046
GETWND	CO 3	WORK	C	NEWSRC	100046
GETWND	SU 5	101 RWELMS		NEWSRC	100046
CKWIND	SU 1	CKWIND		NEWSRC	100047
MKRGF	SU 1	MKRGF		NEWSRC	100048
MKRGF	CO 3	MAFCOM	M	NEWSRC	100048
MKRGF	CO 3	AFIL	C	NEWSRC	100048
MKRGF	CO 3	MAFCOM	M	NEWSRC	100048
MKRGF	CO 3	ELMDAT	C	NEWSRC	100048
MKRGF	CO 3	MAFCOM	M	NEWSRC	100048
MKRGF	CO 3	SIZE	C	NEWSRC	100048
MKRGF	CO 3	MAFCOM	M	NEWSRC	100048
MKRGF	CO 3	CNTLFL	C	NEWSRC	100048
MKRGF	SU 5	101 RWNODS		NEWSRC	100048
MKRGF	SU 5	103 EMINMX		NEWSRC	100048
MKRGF	SU 5	104 RWELMS		NEWSRC	100048
MKRGF	SU 5	105 PROFAC		NEWSRC	100048
MKRGF	RD 7	YES		NEWSRC	100048
MKRGF	WR 8	YES		NEWSRC	100048
PROFAC	SU 1	PROFAC		NEWSRC	100049
PROFAC	CO 3	MAFCOM	M	NEWSRC	100049
PROFAC	CO 3	AFIL	C	NEWSRC	100049
PROFAC	CO 3	MAFCOM	M	NEWSRC	100049
PROFAC	CO 3	ELMDAT	C	NEWSRC	100049
PROFAC	CO 3	MAFCOM	M	NEWSRC	100049
PROFAC	CO 3	SIZE	C	NEWSRC	100049
PROFAC	SU 5	101 RWELMS		NEWSRC	100049
PROFAC	RD 7	YES		NEWSRC	100049
PROFAC	WR 8	YES		NEWSRC	100049
SKIP	SU 1	SKIP		NEWSRC	100050
SKIP	CO 3	MAFCOM	M	NEWSRC	100050
SKIP	CO 3	SIZE	C	NEWSRC	100050
SKIP	RD 7	YES		NEWSRC	100050
HTFIO	SU 1	HTFIO		NEWSRC	100051
HTFIO	RD 7	YES		NEWSRC	100051
RDCNTL	SU 1	RDCNTL		NEWSRC	100052
RDCNTL	CO 3	MAFCOM	M	NEWSRC	100052
RDCNTL	CO 3	AFIL	C	NEWSRC	100052
RDCNTL	CO 3	MAFCOM	M	NEWSRC	100052
RDCNTL	CO 3	CNTLFL	C	NEWSRC	100052
RDCNTL	CO 3	MAFCOM	M	NEWSRC	100052
RDCNTL	CO 3	CNTLTM	C	NEWSRC	100052
RDCNTL	CO 3	MAFCOM	M	NEWSRC	100052
RDCNTL	CO 3	SIZE	C	NEWSRC	100052
RDCNTL	SU 5	101 HTFIO		NEWSRC	100052
RDCNTL	SU 5	102 ELTRAN		NEWSRC	100052
RDCNTL	RD 7	YES		NEWSRC	100052
RDCNTL	WR 8	YES		NEWSRC	100052
TFLSET	SU 1	TFLSET		NEWSRC	100053

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 6

TFLSET	CO 3	MAFCOM	M	NEWSRC	100053
TFLSET	CO 3	AFIL	C	NEWSRC	100053
TFLSET	CO 3	MAFCOM	M	NEWSRC	100053
TFLSET	CO 3	CNTLTM	C	NEWSRC	100053
TFLSET	SU 5 101	SKIP		NEWSRC	100053
TFLSET	RD 7	YES		NEWSRC	100053
GENTMP	SU 1	GENTMP		NEWSRC	100054
GENTMP	CO 3	MAFCOM	M	NEWSRC	100054
GENTMP	CO 3	MAFCOM	M	NEWSRC	100054
GENTMP	CO 3	AFIL	C	NEWSRC	100054
GENTMP	CO 3	MAFCOM	M	NEWSRC	100054
GENTMP	CO 3	SIZE	C	NEWSRC	100054
GENTMP	CO 3	MAFCOM	M	NEWSRC	100054
GENTMP	CO 3	WORK	C	NEWSRC	100054
GENTMP	CO 3	MAFCOM	M	NEWSRC	100054
GENTMP	CO 3	CNTLTM	C	NEWSRC	100054
GENTMP	CO 3	CNTLFL	C	NEWSRC	100054
GENTMP	SU 5 101	TFLSET		NEWSRC	100054
GENTMP	SU 5 102	GETMPE		NEWSRC	100054
GENTMP	SU 5 103	SETFLP		NEWSRC	100054
GENTMP	SU 5 104	RDTEMP		NEWSRC	100054
GENTMP	SU 5 105	CWCRNT		NEWSRC	100054
GENTMP	SU 5 106	ETCORE		NEWSRC	100054
GENTMP	SU 5 107	NTCORE		NEWSRC	100054
GENTMP	SU 5 108	STSTMP		NEWSRC	100054
STSTMP	SU 1	STSTMP		NEWSRC	100055
STSTMP	CO 3	MAFCOM	M	NEWSRC	100055
STSTMP	CO 3	MAFCOM	M	NEWSRC	100055
STSTMP	CO 3	AFIL	C	NEWSRC	100055
STSTMP	CO 3	MAFCOM	M	NEWSRC	100055
STSTMP	CO 3	SIZE	C	NEWSRC	100055
STSTMP	CO 3	SPDAT	C	NEWSRC	100055
STSTMP	SU 5 101	RWSPCF		NEWSRC	100055
STSTMP	SU 5 102	WRSTMP		NEWSRC	100055
WRSTMP	SU 1	WRSTMP		NEWSRC	100056
WRSTMP	CO 3	MAFCOM	M	NEWSRC	100056
WRSTMP	CO 3	MAFCOM	M	NEWSRC	100056
WRSTMP	CO 3	AFIL	C	NEWSRC	100056
WRSTMP	WR 8	YES		NEWSRC	100056
ETCORE	SU 1	ETCORE		NEWSRC	100057
ETCORE	CO 3	MAFCOM	M	NEWSRC	100057
ETCORE	CO 3	MAFCOM	M	NEWSRC	100057
ETCORE	CO 3	AFIL	C	NEWSRC	100057
ETCORE	CO 3	MAFCOM	M	NEWSRC	100057
ETCORE	CO 3	SIZE	C	NEWSRC	100057
ETCORE	SU 5 101	FILREW		NEWSRC	100057
ETCORE	RD 7	YES		NEWSRC	100057
NTCORE	SU 1	NTCORE		NEWSRC	100058
NTCORE	CO 3	MAFCOM	M	NEWSRC	100058
NTCORE	CO 3	MAFCOM	M	NEWSRC	100058
NTCORE	CO 3	AFIL	C	NEWSRC	100058
NTCORE	CO 3	MAFCOM	M	NEWSRC	100058
NTCORE	CO 3	SIZE	C	NEWSRC	100058
NTCORE	SU 5 101	HTFIO		NEWSRC	100058
WRCRNT	SU 1	WRCRNT		NEWSRC	100059
WRCRNT	CO 3	MAFCOM	M	NEWSRC	100059
WRCRNT	CO 3	MAFCOM	M	NEWSRC	100059
WRCRNT	CO 3	AFIL	C	NEWSRC	100059
WRCRNT	WR 8	YES		NEWSRC	100059
HTFPAS	SU 1	HTFPAS		NEWSRC	100060
HTFPAS	CO 3	MAFCOM	M	NEWSRC	100060
HTFPAS	CO 3	SIZE	C	NEWSRC	100060
HTFPAS	RD 7	YES		NEWSRC	100060
SORT	SU 1	SORT		NEWSRC	100061
HTICON	SU 1	HTICON		NEWSRC	100062
HTICON	CO 3	AFIL	C	NEWSRC	100062
HTICON	SU 5 101	SINTIT		NEWSRC	100062

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 7

HTICON	SU 5 102	MARCRD		NEWSRC	100062
HTICON	WR 8	YES		NEWSRC	100062
MARCRD	SU 1	MARCRD		NEWSRC	100063
MARCRD	CO 3	AFIL	C	NEWSRC	100063
MARCRD	SU 5 101	READR		NEWSRC	100063
MARCRD	SU 5 103	SKPREC		NEWSRC	100063
MARCRD	WR 8	YES		NEWSRC	100063
READR	SU 1	READR		NEWSRC	100064
READR	CO 3	AFIL	C	NEWSRC	100064
READR	RD 7	YES		NEWSRC	100064
READR	WR 8	YES		NEWSRC	100064
SKPREC	SU 1	SKPREC		NEWSRC	100065
SKPREC	CO 3	AFIL	C	NEWSRC	100065
SKPREC	RD 7	YES		NEWSRC	100065
SKPREC	WR 8	YES		NEWSRC	100065
SINTIT	SU 1	SINTIT		NEWSRC	100066
SINTIT	CO 3	MAFCOM	M	NEWSRC	100066
SINTIT	CO 3	WORK	C	NEWSRC	100066
SINTIT	CO 3	MAFCOM	M	NEWSRC	100066
SINTIT	CO 3	AFIL	C	NEWSRC	100066
SINTIT	CO 3	KNTDAT	C	NEWSRC	100066
SINTIT	SU 5 101	DATINT		NEWSRC	100066
SINTIT	SU 5 102	PRCSOF		NEWSRC	100066
SINTIT	SU 5 103	PRCGEO		NEWSRC	100066
SINTIT	SU 5 104	WRTNTF		NEWSRC	100066
DATINT	SU 1	DATINT		NEWSRC	100067
DATINT	CO 3	MAFCOM	M	NEWSRC	100067
DATINT	CO 3	WORK	C	NEWSRC	100067
DATINT	CO 3	MAFCOM	M	NEWSRC	100067
DATINT	CO 3	AFIL	C	NEWSRC	100067
DATINT	CO 3	KNTDAT	C	NEWSRC	100067
PRCSOF	SU 1	PRCSOF		NEWSRC	100068
PRCSOF	CO 3	MAFCOM	M	NEWSRC	100068
PRCSOF	CO 3	AFIL	C	NEWSRC	100068
PRCSOF	SU 5 101	GETEWT		NEWSRC	100068
PRCSOF	SU 5 102	GTNUMT		NEWSRC	100068
GETEWT	SU 1	GETEWT		NEWSRC	100069
GETEWT	CO 3	MAFCOM	M	NEWSRC	100069
GETEWT	CO 3	WORK	C	NEWSRC	100069
GETEWT	CO 3	MAFCOM	M	NEWSRC	100069
GETEWT	CO 3	AFIL	C	NEWSRC	100069
GETEWT	CO 3	KNTDAT	C	NEWSRC	100069
GETEWT	SU 5 101	ERRPRT		NEWSRC	100069
GETEWT	RD 7	YES		NEWSRC	100069
GTNUMT	SU 1	GTNUMT		NEWSRC	100070
GTNUMT	CO 3	MAFCOM	M	NEWSRC	100070
GTNUMT	CO 3	WORK	C	NEWSRC	100070
GTNUMT	CO 3	MAFCOM	M	NEWSRC	100070
GTNUMT	CO 3	AFIL	C	NEWSRC	100070
GTNUMT	CO 3	KNTDAT	C	NEWSRC	100070
GTNUMT	SU 5 101	ERRPRT		NEWSRC	100070
GTNUMT	RD 7	YES		NEWSRC	100070
PRCGEO	SU 1	PRCGEO		NEWSRC	100071
PRCGEO	CO 3	MAFCOM	M	NEWSRC	100071
PRCGEO	CO 3	AFIL	C	NEWSRC	100071
PRCGEO	CO 3	KNTDAT	C	NEWSRC	100071
PRCGEO	SU 5 101	ERRPRT		NEWSRC	100071
PRCGEO	RD 7	YES		NEWSRC	100071
WRTNTF	SU 1	WRTNTF		NEWSRC	100072
WRTNTF	CO 3	MAFCOM	M	NEWSRC	100072
WRTNTF	CO 3	WORK	C	NEWSRC	100072
WRTNTF	CO 3	MAFCOM	M	NEWSRC	100072
WRTNTF	CO 3	AFIL	C	NEWSRC	100072
WRTNTF	CO 3	KNTDAT	C	NEWSRC	100072
WRTNTF	SU 5 101	WRTDIR		NEWSRC	100072
WRTNTF	SU 5 102	WRTGEO		NEWSRC	100072
WRTNTF	SU 5 103	WRTEMP		NEWSRC	100072

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 8

WRTNTF	SU 5	104	ERRPRT		NEWSRC	100072
WRTNTF	RD 7		YES		NEWSRC	100072
WRTDIR	SU 1		WRTDIR		NEWSRC	100073
WRTDIR	CO 3		MAFCOM	M	NEWSRC	100073
WRTDIR	CO 3		WORK	C	NEWSRC	100073
WRTDIR	CO 3		MAFCOM	M	NEWSRC	100073
WRTDIR	CO 3		AFIL	C	NEWSRC	100073
WRTDIR	CO 3		KNTDAT	C	NEWSRC	100073
WRTDIR	SU 5	101	ERRPRT		NEWSRC	100073
WRTDIR	RD 7		YES		NEWSRC	100073
WRTDIR	WR 8		YES		NEWSRC	100073
WRTGEO	SU 1		WRTGEO		NEWSRC	100074
WRTGEO	CO 3		MAFCOM	M	NEWSRC	100074
WRTGEO	CO 3		WORK	C	NEWSRC	100074
WRTGEO	CO 3		MAFCOM	M	NEWSRC	100074
WRTGEO	CO 3		AFIL	C	NEWSRC	100074
WRTGEO	CO 3		KNTDAT	C	NEWSRC	100074
WRTGEO	SU 5	101	ERRPRT		NEWSRC	100074
WRTGEO	RD 7		YES		NEWSRC	100074
WRTGEO	WR 8		YES		NEWSRC	100074
WRTEMP	SU 1		WRTEMP		NEWSRC	100075
WRTEMP	CO 3		MAFCOM	M	NEWSRC	100075
WRTEMP	CO 3		WORK	C	NEWSRC	100075
WRTEMP	CO 3		MAFCOM	M	NEWSRC	100075
WRTEMP	CO 3		AFIL	C	NEWSRC	100075
WRTEMP	CO 3		KNTDAT	C	NEWSRC	100075
WRTEMP	SU 5	101	ERRPRT		NEWSRC	100075
WRTEMP	RD 7		YES		NEWSRC	100075
WRTEMP	WR 8		YES		NEWSRC	100075
ERRPRT	SU 1		ERRPRT		NEWSRC	100076
ERRPRT	CO 3		MAFCOM	M	NEWSRC	100076
ERRPRT	CO 3		AFIL	C	NEWSRC	100076
ERRPRT	WR 8		YES		NEWSRC	100076
TEMFOR	SU 1		TEMFOR		NEWSRC	100077
TEMFOR	CO 3		MAFCOM	M	NEWSRC	100077
TEMFOR	CO 3		AFIL	C	NEWSRC	100077
TEMFOR	CO 3		MAFCOM	M	NEWSRC	100077
TEMFOR	CO 3		SIZE	C	NEWSRC	100077
TEMFOR	SU 5	101	TPNAST		NEWSRC	100077
TEMFOR	SU 5	102	MARCTO		NEWSRC	100077
TEMFOR	WR 8		YES		NEWSRC	100077
TPNAST	SU 1		TPNAST		NEWSRC	100078
TPNAST	CO 3		AFIL	C	NEWSRC	100078
TPNAST	RD 7		YES		NEWSRC	100078
TPNAST	WR 8		YES		NEWSRC	100078
I3DSF	SU 1		I3DSF		NEWSRC	100079
I3DSF	CO 3		MAFCOM	M	NEWSRC	100079
I3DSF	CO 3		AFIL	C	NEWSRC	100079
I3DSF	SU 5	101	REAARY		NEWSRC	100079
I3DSF	SU 5	102	SHAPFV		NEWSRC	100079
I3DSF	SU 5	103	CHKPT		NEWSRC	100079
I3DSF	SU 5	104	JACBCK		NEWSRC	100079
I3DSF	SU 5	106	SFDIST		NEWSRC	100079
REAARY	SU 1		REAARY		NEWSRC	100080
REAARY	SU 5	101	SORT2		NEWSRC	100080
SORT2	SU 1		SORT2		NEWSRC	100081
SFDIST	SU 1		SFDIST		NEWSRC	100082
SFDIST	SU 5	101	SHAPFV		NEWSRC	100082
CHKPT	SU 1		CHKPT		NEWSRC	100083
PDOFSF	SU 1		PDOFSF		NEWSRC	100084
PDOFSF	CO 3		CRNDEF	C	NEWSRC	100084
FNDJAC	SU 1		FNDJAC		NEWSRC	100085
SHAPFV	SU 1		SHAPFV		NEWSRC	100086
JACBCK	SU 1		JACBCK		NEWSRC	100087
JACBCK	SU 5	101	PDOFSF		NEWSRC	100087
JACBCK	SU 5	102	FNDJAC		NEWSRC	100087
RWSPCF	SU 1		RWSPCF		NEWSRC	100088

201

TITLE=

DATE =07/15/85

TIME = 23 58

PAGE= 9

RWSPCF	CO 3	MAFCOM	M	NEWSRC	100088
RWSPCF	CO 3	AFIL	C	NEWSRC	100088
RWSPCF	RD 7	YES		NEWSRC	100088
RWSPCF	WR 8	YES		NEWSRC	100088
ELTRAN	SU 1	ELTRAN		NEWSRC	100089
ELTRAN	CO 3	MAFCOM	M	NEWSRC	100089
ELTRAN	CO 3	AFIL	C	NEWSRC	100089
ELTRAN	CO 3	MAFCOM	M	NEWSRC	100089
ELTRAN	CO 3	CNTLFL	C	NEWSRC	100089
ELTRAN	WR 8	YES		NEWSRC	100089
INITCM	SU 1	INITCM		NEWSRC	100090
INITCM	CO 3	CRNDEF	C	NEWSRC	100090
MARCTO	SU 1	MARCTO		NEWSRC	100091
MARCTO	CO 3	MAFCOM	M	NEWSRC	100091
MARCTO	CO 3	AFIL	C	NEWSRC	100091
MARCTO	CO 3	MAFCOM	M	NEWSRC	100091
MARCTO	CO 3	WORK	C	NEWSRC	100091
MARCTO	SU 5 101	GPCOEF		NEWSRC	100091
MARCTO	SU 5 102	GPTEMP		NEWSRC	100091
MARCTO	RD 7	YES		NEWSRC	100091
MARCTO	WR 8	YES		NEWSRC	100091
GPCOEF	SU 1	GPCOEF		NEWSRC	100092
GPTEMP	SU 1	GPTEMP		NEWSRC	100093

REFERENCES

1. "Survey of Computer Programs For Heat Transfer Analysis"

A.R. Noor

2. 3D TITAN User's Manual

R.J. Maffeo

3. PATRAN

PDA Engineering

Software Products Division

1560 Brookhollow Drive

Santa Ana, California 92705-5475

End of Document